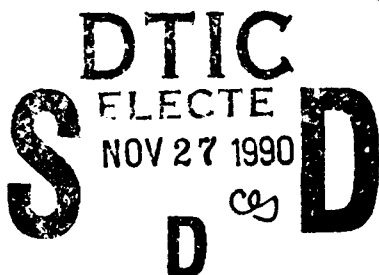


AD-A229 397

David Taylor Research Center

Bethesda, MD 2084-5000

DTRC-90/034 October 1990

Computation, Mathematics and Logistics Department
Research and Development Report**The Integration of the Air Force Content
Data Model and MIL-STD-1388-2B**by
Ruey Chen

DTRC-90/034 The Integration of the Air Force Content Data Model and MIL-STD-1388-2B



Approved for public release; distribution is unlimited.

MAJOR DTRC TECHNICAL COMPONENTS

CODE 011 DIRECTOR OF TECHNOLOGY, PLANS AND ASSESSMENT

12 SHIP SYSTEMS INTEGRATION DEPARTMENT

14 SHIP ELECTROMAGNETIC SIGNATURES DEPARTMENT

15 SHIP HYDROMECHANICS DEPARTMENT

16 AVIATION DEPARTMENT

17 SHIP STRUCTURES AND PROTECTION DEPARTMENT

18 COMPUTATION, MATHEMATICS & LOGISTICS DEPARTMENT

19 SHIP ACOUSTICS DEPARTMENT

27 PROPULSION AND AUXILIARY SYSTEMS DEPARTMENT

28 SHIP MATERIALS ENGINEERING DEPARTMENT

DTRC ISSUES THREE TYPES OF REPORTS:

1. **DTRC reports, a formal series**, contain information of permanent technical value. They carry a consecutive numerical identification regardless of their classification or the originating department.
2. **Departmental reports, a semiformal series**, contain information of a preliminary, temporary, or proprietary nature or of limited interest or significance. They carry a departmental alphanumeric identification.
3. **Technical memoranda, an informal series**, contain technical documentation of limited use and interest. They are primarily working papers intended for internal use. They carry an identifying number which indicates their type and the numerical code of the originating department. Any distribution outside DTRC must be approved by the head of the originating department on a case-by-case basis.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED			1b. RESTRICTIVE MARKINGS	
2a. SECURITY CLASSIFICATION AUTHORITY			3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.	
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE				
4. PERFORMING ORGANIZATION REPORT NUMBER(S) DTRC-90/034			5. MONITORING ORGANIZATION REPORT NUMBER(S)	
6a. NAME OF PERFORMING ORGANIZATION David Taylor Research Center		6b. OFFICE SYMBOL (If applicable) Code 1226	7a. NAME OF MONITORING ORGANIZATION	
6c. ADDRESS (City, State, and ZIP Code) Bethesda, Maryland 20084-5000			7b. ADDRESS (City, State, and ZIP Code)	
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Air Force Human Resources Laboratory		8b. OFFICE SYMBOL (If applicable) AFHRL	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	
8c. ADDRESS (City, State, and ZIP Code) Wright-Patterson Air Force Base Dayton, Ohio 45433-6503			10. SOURCE OF FUNDING NUMBERS	
			PROGRAM ELEMENT NO.	PROJECT NO.
			TASK NO.	WORK UNIT ACCESSION NO.
11. TITLE (Include Security Classification) The Integration of the Air Force Content Data Model and MIL-STD-1388-2B				
12. PERSONAL AUTHOR(S) Chen, Ruey				
13a. TYPE OF REPORT Final	13b. TIME COVERED FROM 1990, Jan TO 1990, Sep	14. DATE OF REPORT (YEAR, MONTH, DAY) 1990, October		15. PAGE COUNT 210
16. SUPPLEMENTARY NOTATION				
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	SUB-GROUP		
19. ABSTRACT (Continue on reverse if necessary and identify by block number) The Air Force is developing a specification for managing technical information concerned with weapon system maintenance and operation. Their aim is to ensure that this technical maintenance information is up-to-date, accurate, readily accessible, and in easy-to-use digital form. The Army is the lead service in implementing the MIL-STD-1388 standard. The implementation of this standard will provide an integrated data base consisting of source data upon which technical maintenance information is based and supported. The benefits of integrating these two closely related data bases are considerable. This report first evaluates these two independently developed data bases and then proposes an integration strategy.				
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT <input type="checkbox"/> DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED	
22a. NAME OF RESPONSIBLE INDIVIDUAL Ruey Chen			22b. TELEPHONE (Include Area Code) (301) 227-1683	22c. OFFICE SYMBOL Code 1226

	Contents	Page
Abstract		1
Administrative Information		1
Executive Summary		2
1 Introduction		3
1.1 Purpose		3
1.2 Scope		3
1.3 Structure of this Report		3
2 Referenced Documents and Publications		4
2.1 Government Documents		4
2.2 Non-Government Publications		6
3 Glossary and Definition of Terms (Also see Appendixes A and B)		7
4 Automation and Integration		7
4.1 Data Dictionary		10
4.2 Conceptual Information Modeling		11
4.3 Logistic Support Analysis Record		13
4.4 Content Data Model		15
5 Data Dictionary of the Content Data Model		19
5.1 CDM Specification (Appendix D)		20
5.2 CDM/NIAM Conceptual Schema (Also see Appendixes E)		20
5.3 CDM/NIAM Data Dictionary (Also see Appendix F)		20
5.4 CDM Neutral Data Model (Also see Appendix G)		21
5.5 CDM/ORACLE-SQL Database Schema (Also see Appendix H)		21
5.6 CDM/Express Data Model (Also see Appendix I)		22
5.7 CDM/IDEF1X Data Model		22



Security Codes	
Dit	Avail and/or Special
A-1	

6	Integration of CDM and MIL-STD-1388-2B	22
6.1	Data Element Level	23
6.1.1	Cross Reference of CDM and MIL-STD-1388-2B Data Elements (Also see Appendix J)	23
6.1.2	MIL-STD-1388-2B Data Elements Available for CDM Use (Also see Appendix K)	23
6.1.3	CDM Data Elements not Available from MIL-STD-1388-2B (also see Appendix L)	23
6.2	Data Schema Level	24
6.2.1	IRDS Standard Approach	25
6.2.2	NIAM Approach	26
6.3	Benefit of the Integration	27
7	Recommendations	27
	Acknowledgements	29

Appendixes

Appendix A - Glossary

10	Glossary	31
----	----------	----

Appendix B - Definition of Terms

20	Definition of Terms	33
----	---------------------	----

Appendix C - NIAM Modeling Guide

30	NIAM Modeling Guide	37
30.1	Objects	37
30.2	Roles	37
30.3	Constraints on Objects and Roles	38

Appendix D - CDM Specification

40	CDM Specification	43
----	-------------------	----

Appendix E - CDM/NIAM Conceptual Schema

50.1	CDM/NIAM Conceptual Schema Descriptions	55
50.2	CDM/NIAM Conceptual Schema Diagrams	61

Appendix F - CDM Data Dictionary

60.1	Scope	79
60.2	Data Element Dictionary Format	79
60.3	Data Element Definition	81

Appendix G - CDM Neutral Data Model Topographical Report

70.1	CDM Neutral Data Model Topographical Report	123
70.2	Cross Reference of the Neutral Data Model	134

Appendix H - CDM/ORACLE-SQL Database Schema

80.1	CDM/ORACLE-SQL Database Schema	139
80.2	Constraints Not Used for ORACLE Database Schema Generation	151

Appendix I - CDM/Express Data Model

90	CDM/Express Data Model	173
----	----------------------------------	-----

Appendix J - Cross Reference of CDM and MIL-STD-1388-2B Data Elements

100	Cross Reference of CDM and MIL-STD-1388-2B Data Elements . .	191
-----	--	-----

Appendix K - MIL-STD-1388-2B Data Elements Available for CDM Use

110	MIL-STD-1388-2B Data Elements Available for CDM Use . . .	195
-----	---	-----

Appendix L - CDM Data Elements Not Available From MIL-STD-1388-2B

120	CDM Data Elements Not Available From MIL-STD-1388-2B . . .	199
-----	--	-----

Figures

Figure 1	- Structure of This Report	5
Figure 2	- Integration of the PDD, LSA and Logistics Databases . .	9
Figure 3	- CDM Generation and Updating	18
Figure 4	- Modes of Technical Manual Development	24

Figure 5 - IRDS Approach of the Integration of the CDM Data Dictionary and a subset of the MIL-STD-1388-2B Data Dictionary . . .	25
Figure 6 - NIAM Approach of the Integration of the CDM Data Dictionary and a subset of the MIL-STD-1388-2B Data Dictionary . . .	26

Tables

Table 1 - LSA Tasks and LSAR tables	14
Table 2 - Levels of Data Dictionary Service	23

Abstract

The Air Force is developing a specification for managing technical information concerned with weapon system maintenance and operation. Their aim is to ensure that this technical maintenance information is up-to-date, accurate, readily accessible, and in easy-to-use digital form.

The Army is the lead service in implementing the MIL-STD-1388 standard. The implementation of this standard will provide an integrated data base consisting of source data upon which technical maintenance information is based and supported

The benefits of integrating these two closely related databases are considerable. This report first evaluates these two independently developed databases and then proposes an integration strategy. (S)

Administrative Information

The Advanced Information Systems Branch (Code 1226) of the Communications and Information Systems Division at the David Taylor Research Center (Carderock) proposed and completed this work. It was sponsored by the Integrated Maintenance Information System (IMIS) program (PE 63106, Project 2950) and managed through the Air Force Human Resources Laboratory. This work was performed under DTRC project number 1820-006, and its DTRC program number was N00167-90-M-6183.

The opinions expressed in this report are those of the author. They do not necessarily reflect the views of the Department of the Navy.

Executive Summary

Computer-aided Acquisition and Logistic Support (CALS) is a joint DoD and industry initiative to facilitate the integration of digital technical information for weapon system acquisition, design, manufacture, and related support functions. Phase II of CALS will fully integrate various independent technical functional capabilities and parallel processes. Their integration will result in considerably enhanced productivity and improved readiness for weapon systems.

In Phase II of CALS, the concept of technical information integration will include the logical collection of shared data models (or physical databases) to support all users of these technical information systems. Data dictionaries which conform to information resources dictionary standards (incomplete at the present time) will be used to produce, manage, and integrate the data models which define the required technical information for weapon systems.

To this end, the Air Force is developing the Content Data Model (CDM) which is a specification for a data base. The CDM is intended to store all of the technical maintenance information for a weapon system in digital form.

The Army is also implementing the MIL-STD-1388-2B standard which defines the requirements of Logistic Support Analysis Record (LSAR) in support of weapon system acquisition and maintenance programs. LSAR is an integrated relational database supporting technical maintenance tasks.

Much of the CDM source data for technical maintenance information should be derived from logistics analysis data such as that provided by LSAR. The integration of LSAR and CDM data will greatly enhance quality and productivity of technical manual development.

Since the CDM specification and MIL-STD-1388-2 were developed independently, they are not compatible with each other. This project was first intended to develop a data dictionary for CDM using the PC-IAST CASE tool which is based on the NIAM binary relationship modeling technique. The PC-IAST tool does not conform at present to the IRDS standard. However, the NIAM method was chosen because it offers a comprehensive set of features suitable for building data models. When this project began in the spring of 1990, there were no CASE tools conforming to the IRDS standard.

The data elements of the CDM and MIL-STD-1388-2B data dictionaries were examined at length. Suggestions for improving compatibility at the data element level are contained in the Appendixes J, K, and L.

Finally, procedures are proposed for developing a prototype integration of a subset of MIL-STD-1388-2B and CDM at the data schema level. The experience gained in integrating these two CALS databases will be crucial in the future development of IWSDB and in planning their integration on a large scale basis, as well as in the development of the IRDS standard itself.

1 Introduction.

For some time, the Air Force has been developing systems and procedures for maintaining technical information for weapon system maintenance and operation to ensure that this information is up-to-date, accurate, and readily accessible in digital form. To this end, the Air Force Human Resources Laboratory (AFHRL) is developing the Content Data Model (CDM) (see Sections 2.1.g and 2.1.j) which is a specification for a database. The CDM defines the logical structure for a revisable and format-free database of integrated technical maintenance information.

The Army is the lead service in implementing the standard MIL-STD-1388-1, requirements for a Logistic Support Analysis (LSA) (see Section 2.1.a) task, and standard MIL-STD-1388-2, requirements for a Logistic Support Analysis Record (LSAR) (see Section 2.1.b) in support of weapon system acquisition and maintenance programs. The resulting LSAR is a logically integrated data base consisting of source data upon which technical maintenance tasks are based and supported.

Clearly, much of the source data of CDM for technical manual (TM) development should be derived from logistics analysis data such as that provided by LSAR. It should be relatively easy to realize the linkage between LSAR and CDM data bases. The database integration between LSAR and CDM should greatly enhance quality and productivity of TM or CDM development.

1.1 Purpose.

Since the CDM specification and MIL-STD-1388-2 were developed independently, they are not compatible with each other at the present time. The purpose of this project is to first analyze these two data standards at the data element level and then to propose a strategy for integrating them.

1.2 Scope.

The scope of this project was first to analyze the CDM by developing a data dictionary and then to assess the compatibility of the CDM data dictionary with that of MIL-STD-1388-2B. Currently the software standards and CASE tools are not mature enough to support a full-scale integration of data dictionaries. This project proposed a strategy for prototyping the integration of a subset of MIL-STD-1388-2B with CDM.

1.3 Structure of This Report.

Figure 1 outlines the structure of this report. Sections 1 through 4 and Appendixes A, B, and C are introductory. Section 2 lists the documents referenced in this report. Section 3 consists of the Glossaries and Definition of Terms. Section 4 discusses four topics which constitute the basis for this report.

Section 5 is the major section of this report. It develops a data dictionary (CDM/NIAM) for CDM which does not conform to the IRDS standard. The Personal-Computer Information Analysis Support Tool (PC-IAST) (see Section 2.2.a), a proprietary product of the Control Data Corporation which is based on the Nijssen Information Analysis Method (NIAM), was used for the development of the CDM/NIAM conceptual schema (see Appendix E) and the CDM/NIAM data dictionary (see Appendix F). The CDM specification (see Appendix D) was used for defining the input requirements in the development of the CDM/NIAM data dictionary. Appendixes G, H, and I contain outputs generated by the CDM/NIAM data dictionary. More detailed descriptions of Boxes II, III, IV, VIII, IX, X, and XI in Figure 1 can be found in Section 5.

Section 6 and Appendixes J, K, and L describe the assessment of the data elements of both CDM/NIAM and MIL-STD-1388-2B and then propose the prototype integration of these two data dictionaries. The recommendations for the integration of these two data dictionaries are summarized in Section 7.

The style and format of this report follow those of the "CALS Program Implementation Guide" (see Section 2.1.e).

2 Referenced Documents and Publications.

2.1 Government Documents.

DoD Standards

- a MIL-STD-1388-1A Logistic Support Analysis
- b MIL-STD-1388-2B DoD Requirements for a Logistic Support Analysis Record (Draft), 6 June 1990.
- c MIL-STD-1840 Automated Interchange of Technical Information.

DoD Specifications

- d MIL-M-28001 Markup Requirements and Generic Style Specification for Electronic Printed Output and Exchange of Text.

DoD Handbooks

- e MIL-HDBK-59A CALS Program Implementation Guide (Draft)
13 October, 1989

DoD Directives

- f DoDD 5000.39 Acquisition and Management of Integrated Logistic Support for Systems and Equipment.

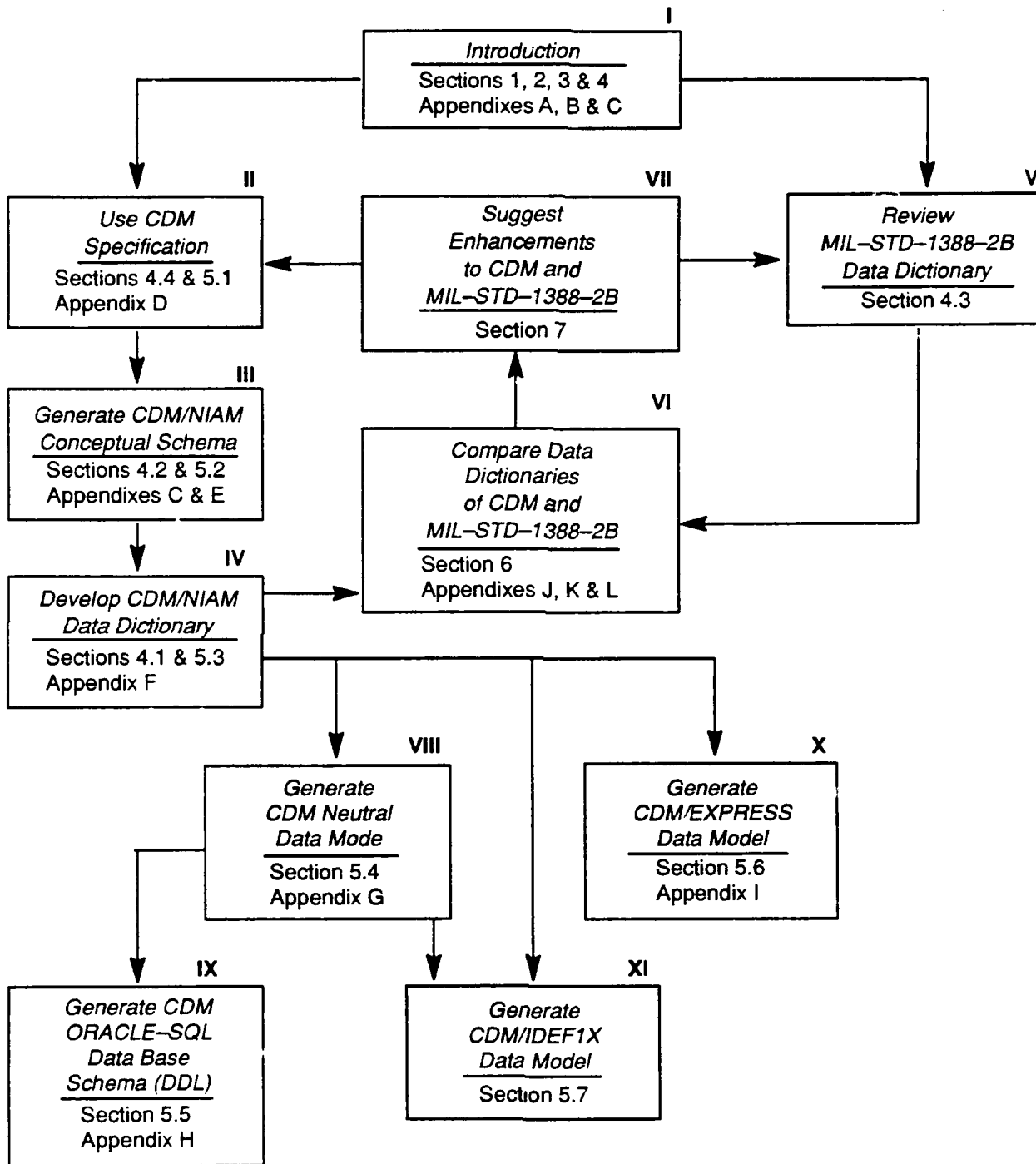


Fig. 1. Structure of this report.

Other Government Publications

- g "Draft Specification for Digital Technical Information", August 1989, Air Force Human Resources Laboratory, Wright-Patterson AFB, OH 45433
- h "Information Modeling Manual - IDEF1X", Air Force ICAM Project, December 31, 1985.
- i Howard Bloom, Cita Furlani, Mary Mitchell, Joan Tyler and Dave Jefferson, "Information Resource Dictionary System: An Integration Mechanism for Product Data Exchange Specification", NISTIR 88-3862, October, 1988.
- j Mark Earl, David R. Gunning, Eric Freese, Ronald Shroder, Walter Werts, and Tim Reser, "Content Data Model: Technical Summary" Air Force Human Resources Laboratory, May, 1990.
- k Alan Goldfine and Patricia Konig, "A Technical Overview of the Information Resource Dictionary System", NBSIR 88-3700, Jan. 1988.
- l David Jefferson, "A Framework for Developing a CALS Data Dictionary", NISTIR 4377, July 1990.
- m Margaret H. Law, "Guide to Information Resource Dictionary System Applications: General Concepts and Strategic Systems Planning", NIST report 500-152, April 1988.
- n Joan Tyler, "A planning Model for Unifying Information Modeling languages for Product Data Exchange Specification", NISTIR 90-4234, September 1989.

2.2 Non-Government Publications.

- a PC-IAST, User's Guide, Control Data Corporation, June 1990.
- b "Briefing to Industry, Canadian Forces Supply System (CFSS) Upgrade Project", Canada National Defense, 22 June 1990.
- c "Information Modeling Language - Express", ISO TC184/SC4/WG1, Document N307, December, 19, 1988.
- d "Information Resource Dictionary System (IRDS) Reference Model" (Draft) ANSI X3H4 Standard Committee, October 1990.
- e "Reference Model for Ship Structural Systems", ISO TC184/SC4/WG1, Document N188, October 17, 1988.
- f "Reference Model for Outfitting System", ISO TC184/SC4/WG1, Document N305, April 11, 1989.
- g "Reference Model for Distribution System", ISO TC184/SC4/WG1, Document N390, August 18, 1989.

- h Martin Bryan, "SGML: An Author's Guide to the Standard Generalized Markup language", Addison-Wesley Publishing Company, 1988.
- i Peter P. Chen, "Proceedings of the International Conference on Entity-Relationship Approach to Systems Analysis and Design", Held December 1979, North-Holland Publishing Co. 1980.
- j Sharad H. Gadre, "Building an Enterprise & Information Model", Database Programming and Design, December 1987.
- k Richard Hull and Roger King, "Semantic Database Modeling: Survey, Applications, and Research Issues", ACM Computing Surveys, Volume 19, No. 3, September, 1987.
- l Mary E. Loomis, "The Database Book", Macmillan Publishing Company, 1987.
- m G. M. Nijssen, T. A. Halpin, "Conceptual Schema and Relational Database Design", Prentice Hall, 1989.
- n Toby J. Teorey, Dongqing Yang and James Fry, "A logical Design Methodology for Relational Databases Using the Extended Entity-Relationship Model", ACM Computing Surveys, Volume 18, No. 2, 1986.
- o Robert I. Winner, James P. Pennell, Harold E. Bertrand and Marko M. G. Slusarczuk, "The Role of Concurrent Engineering in Weapons System Acquisition", Report R-338, Institute for Defense Analyses, December 1988.
- p J. J. Wintraecken, "The NIAM Information Analysis Method: Theory and Practice", Kluwer Academic Publishers, October 1990.

3 Glossary and Definition of Terms.

See "Glossary" used in this report appearing in Appendix A.

See "Definition of Terms" used in this report appearing in Appendix B.

4 Automation and Integration.

The amount of technical and logistical information required to support modern, technologically advanced weapon systems has increased dramatically in recent years. The systems used to originate, process, store, and transfer this technical and logistic information are primarily paper-based systems. In an attempt to diminish the impact of this mass of paper associated with a given weapon system, many selected functions critical to weapon system development and readiness have been automated. These automated support systems are evolving as distinct islands of automation. Though many of these islands of automation may not be compatible, the

benefit of continued investment in automated information systems to reduce paper is obvious. Clearly a greater payoff will be realized if these islands of automation are integrated. This becomes the basis for the DoD and industry joint initiative of the CALS program. The mission of CALS is to support the transition from the current paper-intensive weapon system design, manufacture, and support processes to a largely automated and integrated mode of operation.

CALS implementation is organized into two overlapping phases. Phase I of CALS focuses on converting current paper flows to digital form in a file transfer format (see Section 2.1.c) while Phase II of CALS targets new functional capabilities that will be achieved through integration and consolidation of the numerous parallel or duplicative processes and functions. These efforts will fully integrate logistic support into the product data design process consistent with concurrent engineering practices and will link logistics data products (e. g., technical publications, provisioning documents, and training manuals) with product definition data (PDD). This integration will produce large savings through increased productivity and will result in improved readiness through much improved planning and support. Fig. 2 shows one way to integrate the PDD, LSAR, and logistic databases throughout the weapon system life cycle phases. Most of the PDD and LSAR data for a weapon system is developed in the Full Scale Development phase. The PDD and LSAR data is the primary data source for the logistics data bases development. The feedback obtained from the logistics databases during the system operation phase should be used to enhance the reliability and maintainability of the weapon system.

The CALS Phase II approach to integrating technical data information systems requires an integrated conceptual data model to control and coordinate all the technical information systems supporting a weapon system. This concept of an integrated data model which will include a logical collection of shared data models to support all technical information system users is called the CALS Integrated Weapon System Database (IWSDB) which will provide the basis for the integrated shared data environment. The development of the Product Data Exchange Specification (PDES) will enable information systems to use a standard digital representation (data models) for product data. It will provide a complete, unambiguous, digital definition of physical and functional characteristics of each element/part of a weapon system throughout its life cycle.

To create, integrate, and manage the numerous data models of IWSDB or PDES requires support of a sophisticated standardized information resources management system. Data dictionaries which conform to relevant standards, such as IRDS, should be the mechanism used to acquire weapons system technical data in the CALS Phase II environment. These data dictionaries will be the data management and acquisition vehicle used to define, manage, and produce the required technical data for each weapon system. The initial development of the CALS data dictionary may occur in several major segments: PDD, LSA, and technical maintenance information. Also the initial data dictionaries should be built around the forthcoming release of MIL-STD-1388-2B with respect to defining the logistics related data.

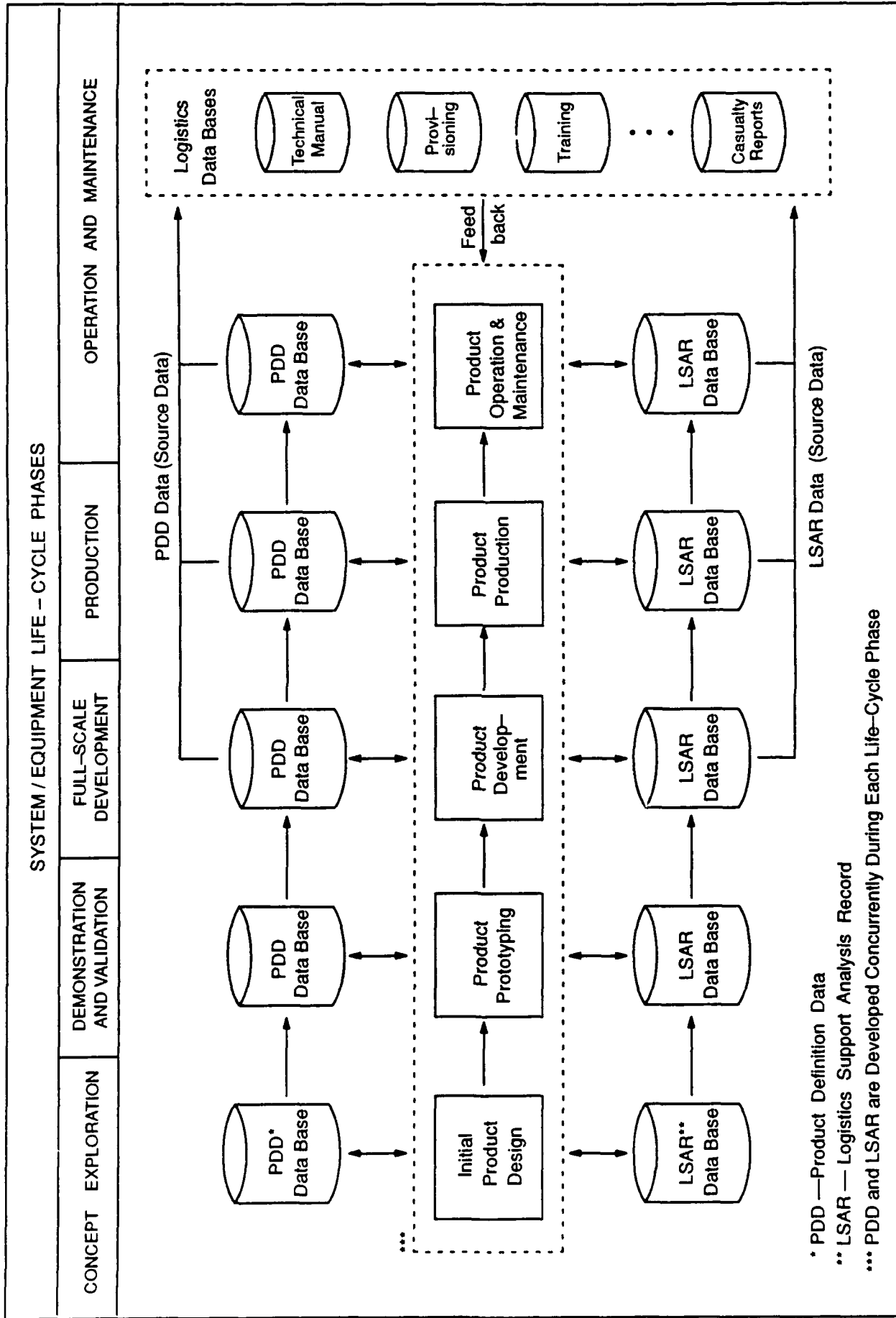


Fig. 2. Integration of the PDD, LSAR, and Logistics data bases.

4.1 Data Dictionary.

A data dictionary is a highly structured type of software system that can be used to model, integrate, and manage data in an organization's information resources. It contains information about the data items it stores; this information about data is called "metadata". Such data dictionaries are increasingly useful integration tools for supporting large information resource management systems such as CALS.

The IRDS is a set of software specifications for a standard dictionary system. An information resource dictionary (IRD) is an IRDS implementation. The IRDS standard was accepted by the American National Standards Institute (ANSI) in 1988 and was published as X3.138-1988. It was also adopted as Federal Information Processing Standard Publication (FIPS PUB) 156.

The purpose of the IRDS standard is to provide users of data dictionary systems with useful, flexible, and user-friendly software products to support all phases of the software system life cycle. The conformance of a data dictionary to the IRDS standard results in greater user efficiency and improved transportability of metadata between systems.

The IRDS is designed to support information modeling. The selection of the conceptual schema representation language in an information system is the crucial factor with regard to the integration of planning, design, software development, data administration, and users in the control architecture. The conceptual schema description of the IRDS standard is currently based on the Entity-Relationship-Attribute (ERA) (see Sections 2.2.i and 2.2.n) data model. The ability of IRDS to accommodate more robust flexible schema description languages such as NIAM must be included.

One of the most important features of the IRDS standard is called "schema extensibility", which means that when new requirements for information resources are defined, the IRD schema can be extended to meet those requirements. This feature also allows the movement of schema or metadata from one IRD to another. If an organization has two or more IRDs, each under the control of an IRDS conforming to the standard, the IRD-IRD transport allows entities and relationships along with their attributes to be transferred from one IRD to another.

Data dictionary systems have permitted the standardization of data elements used by multiple users and facilitated the documentation of such data elements. Data elements are the most basic data structures used in a data dictionary system. As the result of data element standardization, consistent data elements can be efficiently stored in data bases and can also be accessed by multiple users and programs. IRDS supports the standardization of data element names, definitions, and relationships, etc.

The ANSI X3H4 IRDS Standard Committee is completing two additional associated standards, the Export/Import File Format and the Service

Interface, for use in conjunction with the IRDS X3.138-1988 standard. The Export/Import File Format allows the IRDS to exchange data, while the Service Interface provides an interface with the CASE tools. The rectification of these two associated standards is expected to be completed in 1991. However, the current version of the IRDS standard (X3.138-1988) combined with the Export/Import File Format and the Service Interface will not be able to support the integration and life cycle management of the IWSDB/PDES data models developed using the various data modeling languages. The ANSI X3H4 Standard Committee is currently drafting a IRDS Reference Model (Section 2.2.d) which is intended to define the services, interfaces, and architecture of the next version of IRDS which is known as IRDS2. However, unless the X3H4 IRDS standard committee accelerates its development efforts, the completion of IRDS2 specifications is at least four to five years away. The question whether the IRDS2 will support the integration, life cycle management, etc. of IWSDB/PDES depends on the active participation and enthusiastic support of the CALS/PDES communities. The need for such participation and support can not be overemphasized.

Parallel to the data model integration being considered by the X3H4 IRDS committee, the PDES Dictionary/Methodology Committee is developing the PDES Unification Meta Model (PUMM) (Section 2.1.n) which attempts to unify the semantic and data abstraction of various modeling languages. The first three languages targeted by PUMM are Express, IDEF1X, and NIAM. The availability of PUMM for integrating PDES data models is not known at this time.

Recently, a task force from the CALS Industry Steering Group (ISG) was formed to develop plans for the CALS Directory/Data Dictionary Services. Its current charter is to define the scope, requirements, approach, and methodology required for obtaining directory/data dictionary services in a fully integrated CALS environment. Its interim report is expected in November 1990.

4.2 Conceptual Information Modeling.

The most critical aspects of any modern information systems methodology are: specifying the user's information requirements, validating these requirement specifications, and converting them into data base designs. Conceptual data modeling (see Sections 2.2.j, 2.2.l, and 2.2.m) has been used to fulfill these needs. Data modeling is becoming one of the most powerful techniques for establishing and maintaining control over information resources. Data models are used to represent conceptual schemas and to help integrate the information resources.

The input to the data modeling process is the information requirement specifications. The goal of data modeling is to obtain an integrated, formal implementation-independent specification (conceptual schema) of application-specific information. The data model resulting from such data modeling is implementation-independent in the sense that mapping is required to map the conceptual schema into a database management system.

A data model is a rigorous representation of the meaning of data elements and relationships among data elements within some scope of interest. A data model is sometimes referred to as a semantic data model and is typically represented by a graphical technique. If the same technique is used to produce the data models, then these data models considered as a whole can give an overall view of how their respective sets of data fit together. This neutral (conceptual schema) view of data can be mapped to the various physical database structures (internal schemas) as they are implemented and mapped to the user views (external schemas) as they are identified. Integrating large data resources without using data models can be difficult.

Many data modeling techniques are in use today. Data models can be represented in terms of diagrams or language statements. A number of different methodologies (NIAM, IDEF1X, Express, etc.) have been devised for conceptual data modeling. Such a methodology should be capable of supporting concepts such as "entity", "relationship", "cardinality of an entity", etc., as well as describing the requirement specifications captured.

The NIAM methodology (see Sections 2.2.m and 2.2.p) was chosen for CDM modeling because it offers a comprehensive set of features suitable for building data models. The semantic richness and expressive power of NIAM permit the end-user to be fully involved in the design of the information system. It draws the end-user into the data modeling process so that the end-user can verify the model to ensure that the final product will satisfy its requirements.

NIAM attempts to achieve a binary representation of the relationships between object classes. It is sometimes referred as a "binary semantic model". When any meaningful relationship involves more than two entity types, population tables are constructed to ensure the uniqueness and the correctness of the relationship and to avoid anomalies.

The NIAM model is a rich semantic network which emphasizes the meaning of information rather than its structure. NIAM's approach is stronger and more rigid than that of IDEF1X (see Sections 2.1.h, 2.2.1). In particular, mapping from IDEF1X to NIAM is not possible, but mapping from NIAM to IDEF1X is, since mapping from a stronger or equivalent model is always possible. As a result, the quality of the data model will be degraded when the data model schema is mapped from NIAM to IDEF1X.

One of the most interesting features of NIAM is the recognition and graphic portrayal of constraints on relationships. This provides documentation for all the semantics of the relationships between the objects in the scope being modeled. In addition, constraints that cannot be modeled as associative constraints are documented in text.

Perhaps the most useful feature of the NIAM model is its automated derivation of the fifth normal form of the NIAM model. This feature is made possible because NIAM considers the most elementary relationships

between the objects being considered and the constraints that they have to follow. Any further breakdown of the objects would lead to a loss of meaning and referencing ability. Since the objects being modeled represent the lowest level of data entity types, the normalizing algorithm leads to a normalized database schema in the fifth normal form. If a schema suggested by NIAM analysis is found to be not practical, the user has complete freedom to override the NIAM schema and reduce the high level of normality.

The IAST information modeling tool was used to produce the PDES reference models (see Sections 2.2.e, 2.2.f, and 2.2.g). In Canada IAST has been used since 1986 as the information modeling tool for the multimillion dollar effort of upgrading the Canadian Forces Supply System (see Section 2.2.b).

4.3 Logistic Analysis Support Record.

LSA is the implementation of DoDD 5000.39 (see Section 2.1.f). It is the process which combines those management and analysis actions necessary to ensure effective and economical support of a weapon system, both before and after its fielding. The basic management principle of the integrated logistic support process is that logistic support resources must be developed, acquired, tested, and deployed as an integral part of the weapon system acquisition process.

Two functional standards govern LSA and LSAR. The first, MIL-STD-1388-1, defines the LSA process in terms of providing a basis for LSAR. The second, MIL-STD-1388-2, defines those LSAR requirements pertaining to the assembling, managing, and reporting of data. Table 1 depicts the LSA tasks and LSAR records along with a possible strategy for selecting LSA tasks for the weapon system development phases.

The intent of MIL-STD-1388-1 is to ensure reliable, maintainable, and supportable weapon systems at minimum cost by integrating logistic support considerations into the evolving design effort. It is a dynamic, real-time interactive process requiring concurrence of the design, engineering analysis, and product support planning functions. As a result, addressing logistics requirements becomes part of the design process, rather than a consideration after design decisions that excluded support requirements have been made.

MIL-STD-1388-2 defines the format and content of the LSAR and the structure of various standard reports that deliver data in digital form. It consolidates logistics oriented technical information with data from the various engineering disciplines and integrated logistic support elements to reduce redundancy, facilitate timely usage, and promote consistency of data elements from the various disciplines.

DoD is revising MIL-STD-1388-2 and is implementing relational data base technology in the LSAR data system. The draft of revised MIL-STD-1388-2B provides new relational data base tables to replace the LSAR flat files used in MIL-STD-1388-2A. The use of relational

Table 1 – LSA tasks and LSAR tables.

LSA Tasks MIL-STD-1388-2B	LSAR Tables MIL-STD-1388-2B	Possible LSA Strategy for Developmental Items			
		Concep. Explora- tion	Demo./ Valida- tion	Full-Scale Develop- ment	Produc- tion
Management					
100 – Program Planning and Control 101 – Development of an Early LSA Strategy 102 – LSA Plan 103 – Program & Design Reviews		X	X X		
Analysis and Synthesis					
200 – Mission & Support Systems Definition 201 – Use Study 202 – Mission Hardware, Software and Support System Standardization 203 – Comparative Study 204 – Technology Opportunities 205 – Supportability and Related Design Factors	A, X	X X X X X			
300 – Preparation and Evaluation of Alternatives 301 – Functional Requirements Identification 302 – Support System Alternatives 303 – Evaluation of Alternatives and Trade-off Analysis	B, X	X	X X X		
400 – Determination of Logistic Support Resource Requirements 401 – Task Analysis 402 – Early Fielding Analysis 403 – Post Production Support Analysis	C, E, F, G, H, J, U			X X	X X
Test and Correct					
500 – Supportability Assessment 501 – Supportability Test, Evaluation, and Verification			X	X	X

DBMS and tables offers many benefits, among which are ad hoc report generation and a more practical method of on-line access. Since the MIL-STD-1388-2B data base is already logically integrated, the use of other software tools as well as linkage with other related engineering data bases is encouraged. Other significant changes are the incorporation of new data elements, removal of the data entry instructions, and provision of expanded product reports.

The initial draft of MIL-STD-1388-2B was submitted to government and industry for review in May 1989. Review comments indicated that restructuring of the initial draft was needed. The comments were coordinated and implemented in the MIL-STD-1388-2B draft released in August 1990 for second review. The second draft MIL-STD-1388-2B is a vast improvement over the initial draft. However, it is not known when this standard will be released for publication.

4.4 The Content Data Model.

Since weapon systems are becoming more complicated and sophisticated, the weapon system maintenance concept of the 1990's includes numerous diagnostic capabilities, technical and historical information systems, computerized test equipment, and computer aided training. If all these automated support capabilities continue to develop independently, the technicians using them will be required to use several different data systems to access needed information. This problem is also compounded by the massive amount of cumbersome paper-based technical manuals.

To alleviate these problems, AFHRL is developing the Integrated Maintenance Information System (IMIS) to provide technicians with immediate access to all the information needed to efficiently and effectively perform aircraft maintenance. The objective of IMIS is to provide technicians with a rugged, portable computer display system that will interface with other computer systems to provide an integrated database of all the information needed for aircraft maintenance.

When AFHRL considered the requirements for the IMIS, the need for an "integrated, revisable and neutral" database of maintenance information to replace the existing paper-based TM system became apparent. AFHRL proposes CDM (see Sections 2.1.g and 2.1.j) as a data base specification to meet the future needs of weapon system maintenance support and as a standard for interchanging integrated technical maintenance information.

The CDM defines all interrelated maintenance information in terms of content. It is an attempt to consider all such maintenance data as a unified whole, while working toward the goal of integrated maintenance information. The CDM is an integrated database of neutral data in that it does not have embedded information in it that describes how the data is to be used. The main advantage of a neutral data representation is that the information can be used in many different ways by many different application programs. A normalized relational database is a means of representing such neutral data. To ensure that databases of maintenance information created in 1990 remain capable of meeting future needs,

the data representation must be flexible enough to support new uses of the information.

CDM represents technical manual content as a set of discrete "data elements" that are associated via structural "links" that connect related data elements for electronic "hypertext" presentation. For this type of document, separate data elements are linked into a network structure based upon semantic relationships between data elements. For instance, a series of data elements describing the individual steps of a repair procedure might be linked together in this way, with additional linkages from each step to a corresponding illustration, warning, caution, etc. With the help of special presentation software, the user is able to request and access directly the specific information (data elements) needed at a given time.

Each CDM data element has content and attributes as well as relationships to other elements. The classification of the data elements within the CDM can also be expressed using a variety of modeling languages. The CDM is basically a hierarchical structure corresponding to the functional organization of the weapon system. Vehicle information is composed of system information, which in turn is composed of subsystem information, which in turn contains parts and troubleshooting information, as well as a collection of functions which describe maintenance tasks in terms of the steps required to complete them.

The CDM also contains narrative information, warnings and cautions, cross-references, and configuration management attributes. The individual elements within the CDM can be represented in many ways: text, graphics, tables, video frames, audio, etc. Each data element within the CDM is uniquely identified and therefore can be retrieved and manipulated individually. Shared data elements are stored only once. Each data element within the hierarchy can have attributes that identify content type, security, validation status, etc.

The CDM maintains compatibility with MIL-STD-28001 (see Section 2.1.d) by using the SGML (see Section 2.2.h) methodology as the specification language for describing the structure of documents. The resulting CDM specification is realized in an SGML Document Type Definition (DTD). Although the CDM employs SGML, the physical storage of the database it defines should vary from application to application. SGML has a flexible, parsable, and hierarchical syntax. SGML makes a clear distinction between data elements and the formatted representations of those elements.

The CDM is a generic way to structure an assemblage of maintenance information from many different types of documents in such a way that the links are retained and redundancy is held to a minimum. The CDM specifies a data structure which is useful for transferring information from one agency to another, such as, for example, data transferred from a prime contractor to the Government to fulfill a contract data requirement. Under this concept, the contractor prepares and uses a document instance (DI) as a deliverable for a contract. The contract must previously

specify the SGML Declaration and the DTD to be used for the DI. During the preparation of the DI, the contractor is free to use whatever internal authoring tools he feels are productive. Just prior to shipping, the DI is checked against the contractor's copy of SGML Declaration and the DTD.

Figure 3 depicts a life-cycle phase scenario for generating and updating a CDM instance. Figure 3 also shows the relationship between the contractor's source data (CDM and LSAR data), the CDM data (a view of the contractor's source data), and the technical manual instances (views of the CDM data). The PDD (Box I of Figure 3) contains design, engineering, and manufacturing information which can be in the form of text or graphics files. The PDD provides the CDM with all the graphics required for illustrations and diagrams. The format of these graphics can be CGM, IGES, etc. These digital graphics files have been generated and stored in the PDD database, and they can be invoked by the CDM as external files.

The LSAR (Box II) contains all the maintenance support information required by the CDM instance. A CDM instance (Box III) can be generated from the LSAR and PDD databases. This CDM instance contains all the maintenance support information for either a piece of equipment, or one or more weapon systems in the form of text marked up with SGML tags. The CDM.DTD shown above the arrow head line connecting Boxes II and III can be regarded as a template or a filter with respect to this SGML marked up text loaded to the CDM from the LSAR. It should be noted that the arrow head lines of Figure 3 can be considered as processes which process and transfer data contained in one box to another box as indicated.

After the CDM instance (Box III) is generated, instances (SGML markups) of different types of technical manuals or view packages can be generated from the CDM instance by "filtering" it with respect to the DTDs for these types of technical manuals. For example, the CDM contains maintenance support information concerning the Job Guide, Fault Isolation, and Depot Level procedures, etc. If the Job Guide DTD (JG.DTD) shown above the line connecting Boxes III and IV is applied to the CDM, then the Job Guide instance generated (Box IV) is in conformity with the JG.DTD; that is, the Job Guide instance will parse against the JG.DTD. This Job Guide instance will contain required information for the organizational level maintenance of a piece of equipment or a system. Similarly the instance in Box V is geared toward the Fault Isolation DTD (FI.DTD).

The technical manual instances in Boxes IV or V consist of SGML marked up text which does not contain formatting information. As a result, such instances are difficult for people to read. More importantly, these instances can not be processed for output with respect to a medium (paper hard copy, screen display, etc.) unless appropriate formatting information is combined with these instances. Such formatting information is provided by a Formatted Output Specification Instance (FOSI) defined below.

An Output Specification (OS) is an SGML construct which details the formatting options available for the output of technical manual instances

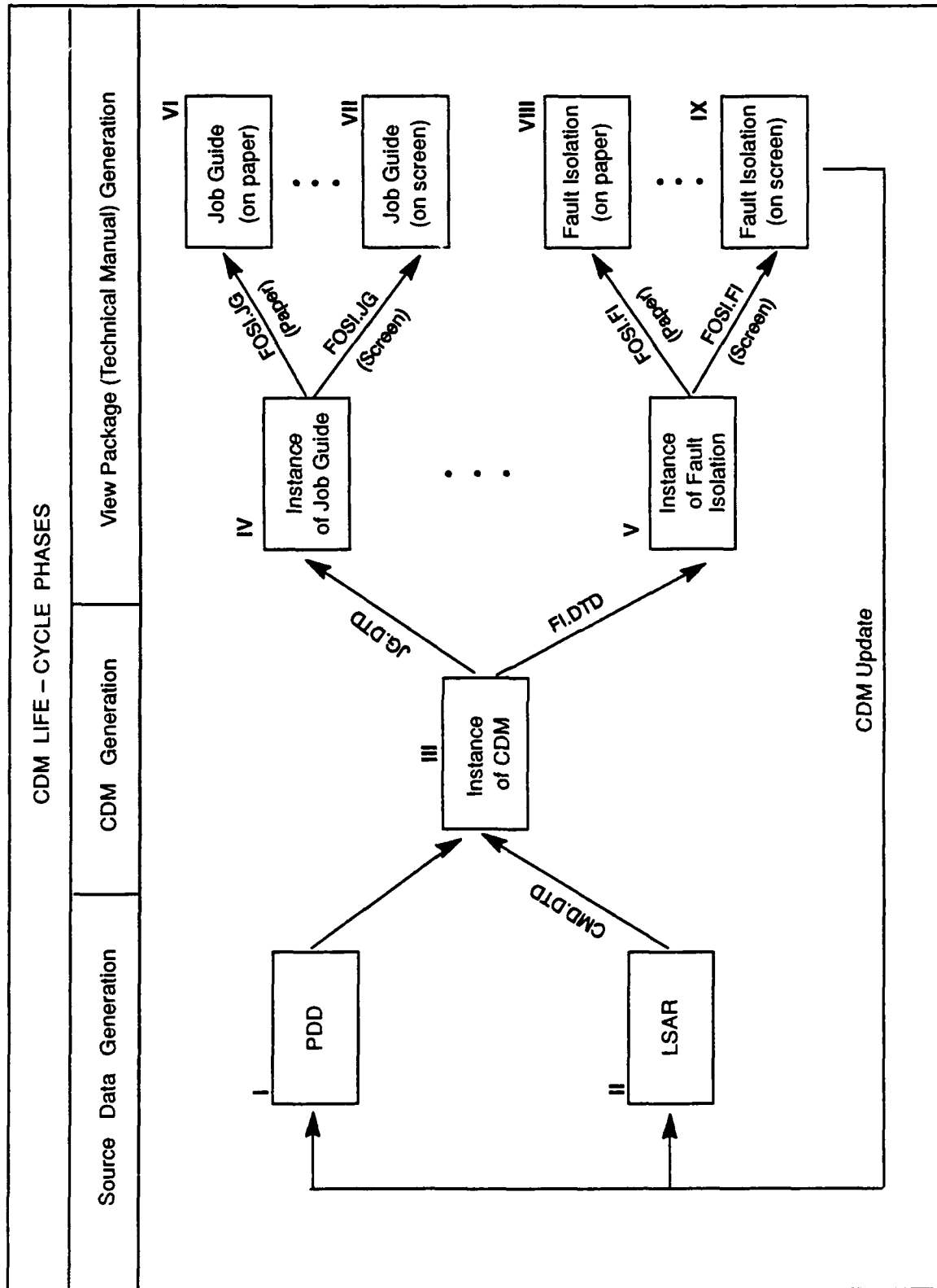


Fig. 3. CDM generation and updating.

conforming to DTDs such as the Job Guide (JG.DTD) and Fault Isolation (FI.DTD), etc. The current version of the OS is geared specifically toward paper documents. In the future, either the OS will be extended to other display media or an OS will be developed for each display medium.

A Formatted Output Specification Instance (FOSI) is an SGML construct which specifies those values chosen for the formatting options from the available values in the OS, and which binds those choices to elements and their attributes of a particular DTD with respect to formatting. The OS must fully support all the characteristics of an output medium while the FOSI merely specifies the chosen values for the relevant options with respect to a given formatting application.

After the FOSIs for the Job Guide technical manual have been obtained from their respective OSs, the Job Guide instance is combined with the FOSI for the chosen medium for output processing. In particular, Box VI contains the paper hard copy of the Job Guide technical manual generated from the combined processing of the Job Guide instance (Box IV) and Job Guide FOSI for paper output. Likewise, Box VII contains the screen display of the Job Guide technical manual generated from the combined processing of the Job Guide instance and the Job Guide FOSI for screen display.

With respect to the Job Guide instance (Box IV), it should be noted that the FOSI for paper hard copy output (Box VI) will be different from the FOSI for screen display output (Box VII). Moreover the FOSI for paper hard copy output (Box VI) will also be different from the FOSI for the paper hard copy output (Box VIII) for the Fault Isolation instance (Box V).

Any changes or updates to the CDM instance (Box III) due to the correction of errors found in the technical manuals or the imposition of new requirements will have to be implemented at the source data level in the PDD (Box I) and the LSAR (Box II).

The CDM instance (Box 3) can be used to instantly generate a CDM view package which includes all necessary maintenance information for a particular maintenance job, and which can be loaded to a portable storage and display device for the use of a maintenance technician.

5 Data Dictionary of Content Data Model. (Also see Appendix F)

The CDM/NIAM data dictionary (Section 60) was developed using PC-IAST (see Section 2.2.a) which is an NIAM implementation. PC-IAST is a CASE tool developed by the Control Data Corporation. Boxes II, III and IV of Figure 1 show the development process for the CDM/NIAM data dictionary. Boxes VIII, IX, X, and XI of Figure 1 represent the various data models and outputs which can be generated from the CDM/NIAM data dictionary.

5.1 CDM Specification. (Also See Appendix D)

Section 40 contains the CDM specification (developed by AFHRL) defining revisable, format-free technical information for weapon system maintenance. These CDM specifications take the form of an SGML DTD which describes the logical structure of a technical information database. These specifications are not intended to describe a database definition. Rather their purpose is to describe the data elements that will make up any acceptable database design.

5.2 CDM/NIAM Conceptual Schema. (Also see Appendix E)

The CDM specification (Section 40) are used as the input requirement for the development of the CDM/NIAM conceptual schema (Section 50.2).

The PC-IAST software tool is employed to graphically identify the objects, relationships of objects, and constraints between objects of the CDM for consistency to ensure that the CDM/NIAM model developed is complete and internally consistent. The CDM/NIAM model is a rich semantic network that concentrates on the meaning of information rather than its structure. For the detailed steps for generating the CDM/NIAM conceptual schema, see the instructions in Section 2.2.a.

PC-IAST provides the ability to create and maintain many distinct data dictionaries or data models. One model may have one or more "scopes" which can be considered as a "user view". PC-IAST may be applied to a model or to a scope of a model. Scope-based operations are useful for obtaining reports and diagrams of a portion of a model for review as well as specifications of a distributed portion of the information base. The CDM/NIAM conceptual schema developed are presented twenty scopes or submodels as shown in Section 50.2.

PC-IAST is capable of producing a graphical plot of a model on a printer or a plotter. The graphics file generated by the GRAFTOOL capability of PC-IAST uses the exchange format "DXF". This graphics file can be used as input to a computer-aided design (CAD) system capable of accepting DXF format files, such as AutoCAD. The CAD system is used to obtain a hard copy of the conceptual schema of a model or a scope of a model. The schema displayed on the screen of the CAD system can also be manipulated and modified. Section 50.2 contains the graphic plots generated by AutoCAD using the CDM/NIAM conceptual schema.

PC-IAST supports the exchange of models between other installations of PC-IAST. The transfer files containing the data dictionary stored on a floppy disk can also be used for archiving or backup purposes.

5.3 CDM/NIAM Data Dictionary. (Also see Appendix F)

Each time a new conceptual schema is generated, it is necessary to create a data dictionary database which is used to design, locate, protect, manage, and control data items with respect to:

- category of data element
- definition of data element
- relationship of the data element to other data items
- format of the data element.

After the completion and validation of the CDM/NIAM conceptual schema (Section 50.2), the generation of the CAM/NIAM data dictionary is complete. This CDM/NIAM data dictionary can be used to generate other data models and data schema which are described in the following four sections.

5.4 CDM Neutral Data Model. (Also see Appendix G)

The Neutral Data Model is a normal form record model generated from the conceptual semantic schema which provides an overview of the data record structure. This Neutral Data Model represents the optimal design of the data structure because it eliminates update anomalies and enables the database to support any valid functional use of data. The schema thus generated can be regarded as a neutral data model that can be introduced into a appropriate relational DBMS. The neutral aspect of the record model means that it is independent of any database structure or product.

PC-IAST offers two grouping modes: "strong" and "weak". The strong model is a fully normalized grouping that preserves the model as defined. No reference class suppression or null valued fields are allowed. The weak grouping makes provision for adjusting the grouping rules to produce a simpler data model by suppressing reference classes and permitting null valued fields.

The topographical print (see Section 70.1) produces the record structures in diagrammatical form, indicating the relative subset hierarchical relationships between the records. Structures which have no subset relationships appear first, followed by records which are subsets of the records displayed initially. The neutral data model diagram contains the structures with all the fields in each structure, as well as the uniqueness constraints that apply to the fields.

A cross-reference list (see Section 70.2) indicates the path number with its "to" and "from" origins. Above each record, the unique identifiers are indicated with the standard double-arrowed lines.

The superset is indicated in the record with the "^" character appearing in the bottom part of the record. A path number under this character "tiers" the superset to its corresponding subset. The subset is indicated by the "<" character appearing in the top part of the record. Its path number appears above this character.

5.5 CDM/ORACLE-SQL Database Schema. (Also see Appendix H)

Once the neutral data model has been generated and reviewed, the next step is to transform the data model into a data definition language for a DBMS. The PC-IAST software supports several different DBMS products. In fact, there are eight "pipes": SQL (ISO SQL86), Oracle, IM/DM,

IM/PERSONAL, INFOCEN, INGRESS, INFORMIX, and ASN.1 (OSI Abstract Syntax Generation) supported by PC-IAST.

Section 80.1 shows an example that an ORACLE database schema has generated. The system produces a report of the constraints (see Section 80.2) which could not be taken into account due to the capabilities currently available in the ORACLE database manager.

5.6 CDM/Express Data Model. (Also see Appendix I)

The CDM/Express data schema (Section 90) is automatically generated by the PC-IAST from the CDM/NIAM data dictionary.

5.7 CDM/IDEF1X Data Model.

PC-IAST can be used to transform the NIAM conceptual schema into data model documentation consistent with the IDEF1X version of the ER modeling technique. PC-IAST will automatically produce documentation and diagrams from an IDEF1X data model generated from the NIAM model and its associated CDM neutral data model.

PC-IAST can generate IDEF1X diagrams with three levels of detail: entity-only, key-only, and full-attribute. The graphic diagrams generated are formatted in DXF which can be plotted on an AutoCAD system.

Since the generation of the CDM/IDEF1X data model is not funded for this project, the PC-IAST tool option for converting NIAM to IDEF1X has not been exercised.

6 Integration of CDM and MIL-STD-1388-2B.

A sophisticated data dictionary system should provide the five levels of services outlined and described in Section 2.1.1. These five levels of service are listed in the first column of Table 2 below. For the purpose of comparison, references to the relevant dictionary data of CDM and MIL-STD-1388-2B are provided in the second and third columns, respectively.

It should be noted that at present the relevant standards as well as the software CASE tools do not support the integration of these two data dictionaries: CDM (using NIAM as data modeling method) and MIL-STD-1388-2B (using IDEF1X as data modeling method). Accordingly, this section first compares the data elements of CDM/NIAM and MIL-STD-1388-2B to assess the compatibility of these two data dictionaries, and then it proposes strategies for integrating a maintenance task related subset of a MIL-STD-1388-2B data dictionary into a CDM data dictionary to form a new data dictionary hereafter called the CDM-LSAR data dictionary.

Table 2 – Levels of Data Dictionary services.

Level of Service		CDM	MIL-STD-1388-2B
1	Define Data Element	Appendix F	Appendix E
2	Define Simple Structure	Appendix F	Appendix A
3	Define Business Rules	Appendix F	
4	Support Multiple Views	Appendix E	
5	Support Life Cycle Management		

6.1 Data Element Level.

When the proposal for this project was developed in the fall of 1989, the MIL-STD-1388-2B was expected to be published in the Spring of 1990, but the publication of this standard was delayed. In August 1990, the second draft of MIL-STD-1388-2B (dated 6 June 1990) was received. This draft version of MIL-STD-1388-2B was used for this report.

The CDM/NIAM data elements contained in Appendix F of this report and the data elements contained in Appendixes A and E of MIL-STD-1388-2B were compared and assessed.

6.1.1 Cross reference of CDM and MIL-STD-1388-2B Data Elements. (See Appendix J)

Section 100 contains matched data elements of CDM and MIL-STD-1388-2B. The rows of Section 100 contain those CDM data elements which can be derived from data elements of MIL-STD-1388-2B. The data definition and data format of the matched data elements may not agree exactly.

6.1.2 MIL-STD-1388-2B Data Elements Available for CDM Use. (See Appendix K)

Section 110 contains those data elements of MIL-STD-1388-2B which can be used as source data and also added to CDM.

6.1.3 CDM Data Elements Not Available From MIL-STD-1388-2B. (See Appendix L)

Section 120 contains a list of CDM data elements for which the source data is not available from MIL-STD-1388-2B.

6.2 Data Schema Level.

The maintenance task support function contained in MIL-STD-1388-2 was originally intended to provide a summary of maintenance task requirements such as those provided in LSAR reports 023 and 024.

The current practice in TM development is to first develop the related LSAR data records and reports and then use them as source or reference data for use in authoring a TM as depicted in Figure 4.

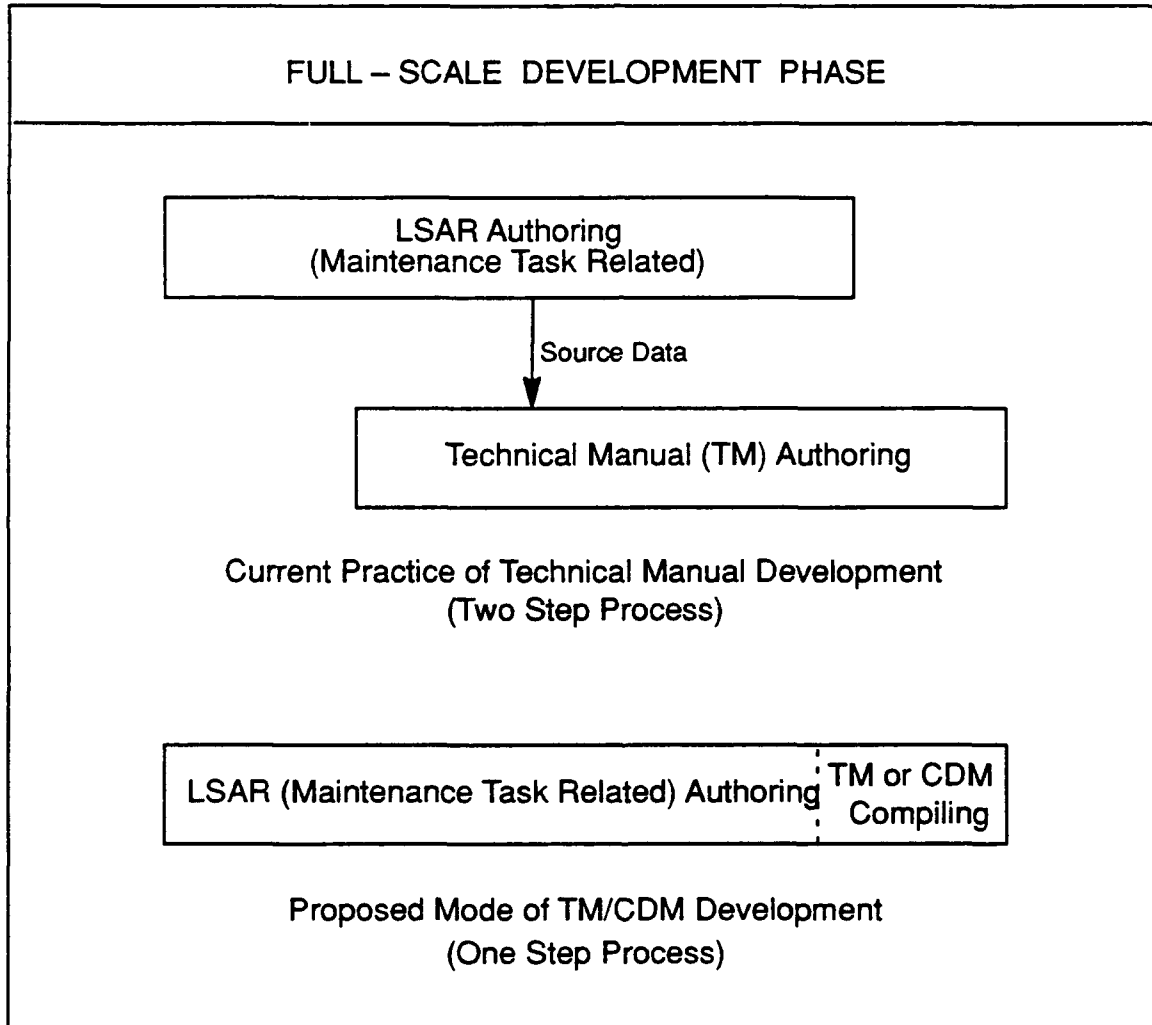


Fig. 4. Modes of technical manual development.

An alternative procedure in TM development is to combine this two step process into a one step process as shown in the lower portion of Figure 4. This combined one step process requires the integration of the LSAR and

CDM databases. In this one step process, all the data needed for the TM development is obtained at the LSAR stage. This one step process can considerably diminish the TM (or CDM) development time and cost for both Government and contractor. In the one step approach, the Government needs to define the TM requirements and go through validation and acceptance procedures for the TM product only once. Likewise industry needs to develop and deliver the TM product only once.

6.2.1 IRDS Standard Approach.

It is proposed that the CDM data dictionary and a maintenance task related subset of the MIL-STD-1388-2B data dictionary be combined to form the new CDM-LSAR data dictionary. IRDS CASE tools will be used for the development work. Figure 5 depicts the generation of this new data dictionary. The integrated CDM-LSAR data dictionary using ERA data modeling language is created in the following way:

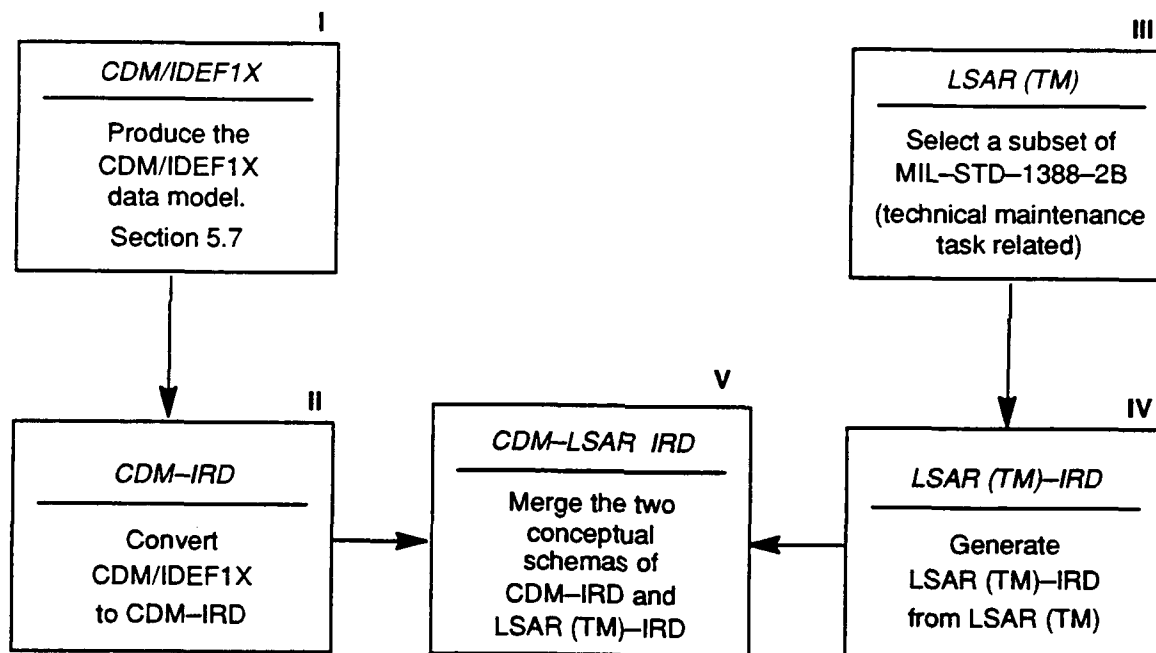


Fig. 5. IRDS approach of the integration of the CDM Data Dictionary and a subset of the MIL-STD-1388-2B Data Dictionary.

a. Use the PC-IAST CASE tool to convert the CDM/NIAM data model of Section 5.7 to a CDM/IDEF1X data model (Box I of Figure 5).

b. Use the IRDS CASE tool to generate CDM-IRD (ERA data model) from the CDM/IDEF1X data model (Box II of Figure 5). Although the IRDS standard

uses the ERA modeling technique, there are no significant differences between the ERA and IDEF1X modeling techniques.

c. Select a technical maintenance task supporting subset from Appendix A of MIL-STD-1388-2B and call it "LSAR(TM)" (Box III of Figure 5).

d. Generate the LSAR(TM)-IRD (ERA data model) with the IRDS CASE tool (Box IV of Figure 5).

e. In Box V, use the IRDS CASE tool to compare the similarities and differences in the definition, structure and format of the data element across the CDM-IRD and LSAR(TM)-IRD data models needed for integration.

6.2.2 NIAM Approach.

The PC-IAST CASE tools can also be used to integrate the CDM and MIL-STD-1388-2B data dictionaries. The procedures would be similar to the one in Section 6.2.1. NIAM would be used instead of the ERA modeling methodology. The CASE tool PC-IAST provides the capabilities required to merge (see Figure 6) the NIAM data models.

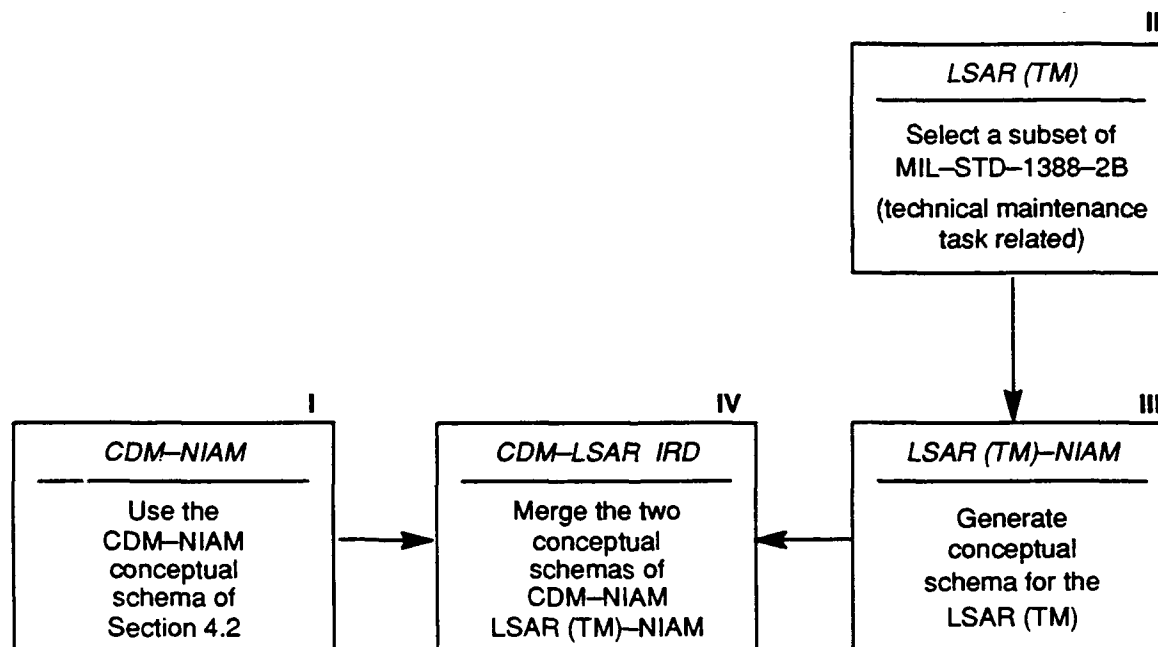


Fig. 6. NIAM approach of the integration of the CDM Data Dictionary and a subset of the MIL-STD-1388-2B Data Dictionary.

The integrated CDM-LSAR data dictionary with NIAM is created in the following way.

- a. Use the conceptual schema of CDM-NIAM created as indicated in Section 4.3. (Box I of Figure 6).
- b. Select a technical maintenance task supporting subset from Appendix A of MIL-STD-1388-2B and call it "LSAR(TM)" (Box II of Figure 6).
- c. Generate the LSAR(TM)-NIAM with the PC-IAST CASE tool (Box III of Figure 6).
- d. Merge these two conceptual schema of CDM-NIAM and LSAR(TM)-NIAM to form a new conceptual schema of CDM-LSAR (Box IV of Figure 6).

6.3 Benefits of the Integration.

The following benefits would be obtained from this prototype integration of CDM and a maintenance task supporting subset of MIL-STD-1388-2B:

- a. The integrated CDM-LSAR data dictionary can be used as the TM authoring database.
- b. The results obtained from the analysis and integration of CDM and MIL-STD-1388-2B will provide valuable feedback for their further integration efforts.
- c. The experience gained in integrating these two CALS data models will be valuable in the design and planning of the future integration of the IWSDB/PDES data models on a larger scale.
- d. The lessons learned in using the IRDS CASE tool in the CALS environment are of great value in the development of the IRDS standard.

7 Recommendations.

Work on this project began in January 1990. When the project had reached the point of writing the final report in August 1990, AFHRL upgraded the CDM specification to Version 5.3. At the same time, the second government and industry review of MIL-STD-1388-2B was under way. Clearly, this project is just the beginning, not the end, of attempts to integrate CDM and MIL-STD-1388-2B.

Since the CDM specification and the MIL-STD-1388-2B standard were developed independently, they are not compatible with each other at the present time. Moreover, because the potential benefit from integrating these two CALS databases is considerable, the following work is recommended for continuation.

- a. Update the CDM Specification.

Since AFHRL upgraded the CDM specification to version 5.3 in August 1990, update the CDM/NIAM data dictionary of Section 5 of this report with the new version of CDM. Also verify the CDM/NIAM conceptual schemas

using the "population table" facility provided by the latest version (August 1990) of PC-IAST. Finally develop the CDM/IDEF1X data model as indicated in Section 5.7.

b. Enhance the compatibility of CDM and MIL-STD-1388-2B at the data element level.

Appendixes J, K, and L of this report should be used as a reference to enhance the CDM specification and the MIL-STD-1388-2B to make them more compatible with each other at the data element level. (see Section 6.1)

c. Develop a prototype of the integration of CDM and a subset of MIL-STD-1388-2B with IRDS standard.

Employ IRDS CASE tools (such as the IRMS developed by InfoSpan) to integrate CDM and a subset of a MIL-STD-1388-2B (technical maintenance task related) data dictionary. For details, see Section 6.2.1.

d. Develop a prototype of the integration of CDM and a subset of MIL-STD-1388 -2B with PC-IAST CASE tool.

As an alternative to the IRDS standard, a PC-IAST CASE tool is available to integrate CDM and a subset of a MIL-STD-1388-2B (technical maintenance task related) data dictionary. For details, see Section 6.2.2.

e. Whenever PUMM (see Section 2.1.n) becomes available for use, prototype the integration of CDM and MIL-STD-1388-2B with PUMM.

f. Actively participate in and support the IRDS standard development by the CALS/PDES communities.

Acknowledgements

Barely a year has passed since the preparation of the proposal for this project. Since then funding for the work was obtained, the work was performed, and this report was prepared. The completion of this project in such a short time was possible only with the help of many people.

The author first wishes to thank his division head William Camara and his branch head Joseph Garner for their technical guidance and assistance. Special appreciation is extended to his colleague Donald Gignac who participated in many hours of technical discussion and who edited this report. In particular his assistance in the preparation of Figure 3 is gratefully acknowledged. The author also thanks his colleague Mary McCarthy for illuminating technical discussions.

The author wishes to acknowledge the assistance of the following DTRC personnel: Robert Schaffran and Jack Brainin for funding the initial study and proposal writing; Joseph Fuller for expediting the transfer of project funds from the Air Force to DTRC; and Patricia Blake for typing this report; Carol Naas for preparing the figures and tables for publication; and Alice Phillips for reviewing the final draft.

The Author wishes to thank the following AFHRL personnel: David Gunning for both funding this project, and providing his technical assistance when required; and Lt. Brian Caporlette for his technical assistance.

Finally, the author wishes to extend his thanks to the following people in private industry: Pam Gennusa and James Oldfield of Datalogics for their review and comments on the draft of Figure 3; Paul Thompson of Control Data Corporation (CDC) and John Zimmerman of Allied Signal for their review and comments on the draft of Appendix C; Joan Milloy of CDC for her assistance in compiling Appendix E; and Ron Shroder and Walter Werts of RJO for their technical assistance.

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix A - Glossary

10 Glossary

AFHRL	Air Force Human Resources Laboratory.
ASN.1	Abstract Syntax Generation.
AutoCAD	A proprietary computer aided design software engineering tool.
CAD	Computer aided design.
CALS	Computer-aided Acquisition and Logistic Support.
CASE	Computer Aided Software Engineering.
CDM	Content Data Model.
DBMS	Data Base Management System.
DDL	Data Definition Language.
DI	Document Instance.
DoD	Department of Defense.
DoDD	Department of Defense Directive.
DTD	Document Type Definition.
DXF	Data Exchange Format.
ERA	Entity Relationship Attribute.
FI	Fault Isolation.
FIPS	Federal Information Process Standard.
FOSI	Formatted Output Specification Instant.
FSD	Full Size Development.
GRAFTOOL	A function of PC-IAST.
HDBK	Hand book.
IAST	Information Analysis Support Tool.
IDEF1X	ICAM Definition Extended.
IMIS	Integrated Maintenance Information System.

INFOCEN A proprietary database management system.
 INFORMIX A proprietary database management system.
 INGRESS A proprietary database management system.
 IRD Information Resource Dictionary.
 IRDS Information Resource Dictionary System.
 IRMS InfoSpan Resource Management System, a proprietary IRDS CASE tool.
 ISO International Standard Organization.
 IWSDB Integrated Weapon System Database.
 JG Job guide.
 LSA Logistic Support Analysis.
 LSAR Logistic Support Analysis Record.
 MIL Military.
 NIAM Nijssen Information Analysis Method.
 NIST National Institute of Standards and Technology
 OS Output Specification.
 PDES Product Data Exchange System.
 PDD Product Definition Data.
 SGML Standard of General Markup Language.
 SQL Structured Query Language.
 STD Standard.
 TM Technical Manual.

Appendix B - Definition of Terms.

20 Definition of Terms.

Attribute - A characteristic or descriptor name of an entity.

Cardinality - The number of instances of one entity that can be related to an instance of another entity.

Computer-aided Acquisition and Logistics Support (CALS) - A joint DoD and industry initiative to facilitate the integration of digital technical information for weapon system acquisition, design, manufacture, and support functions. For details, see Sections 2.1.c and 2.1.e.

Concurrent Engineering - A systematic approach to creating a product design that considers all elements of a product's life-cycle from conception through disposal. For details, see Section 2.2.o.

Content Data Model (CDM) - A specification that defines a revisable, format-free database of integrated technical information. For details, see Sections 2.1.g and 2.1.j.

Data Dictionary - A type of database, containing metadata used to design, monitor, document, and control data in information resource systems. For details, see Sections 2.1.k and 2.1.m.

Data Model - A data model is a rigorous representation of of data elements and the relationships among them with respect to some unifying factor or factors. For details, see Section 2.2.k.

Document Instance (DI) - Text marked up with SGML tags with respect to a particular DTD. For details, see Section 2.1.d.

Document Type Definition (DTD) - An SGML construct which completely defines and unambiguously describes the structure of a class of documents. For details, see Section 2.1.d.

Entity - Refers to either of the following:

- o Entity Instance - A specific object, either real or abstract, used to store data.
- o Entity type - A collection of entities that have similar properties.

Entity-Relationship-Attribute (ERA) Model - An extended version of Entity-Relationship (ER) Model (see Section 2.2.i). It is a methodology or technique for data modeling. For details, see Section 2.2.n.

Express - A PDES language used to model information. For details, see Section 2.2.c.

Formatted Output Specification Instance (FOSI) - An SGML construct

which specifies the formatting options for the document instance with respect to an Output Specification. For details, see Section 2.1.d.

Information Analysis Support Tool (IAST) - A CASE tool for information modeling based on NIAM. For details, see Sections 2.2.a.

Integrated Computer-Aided Manufacturing Definition (IDEF1X) - A methodology used to produce a data model which represents the data structure and semantics of an information system. For details, see Sections 2.1.h and 2.2.1.

Information Resource Dictionary (IRD) - A data dictionary application managed by an IRDS.

Information Resource Dictionary System (IRDS) - A set of standards and specifications resulting from both Federal and international efforts for data dictionary systems. For details, see Sections 2.1.k. and 2.1.m.

Logistic Support Analysis (LSA) - Analysis concerning the scope and level of logistic support. For details, see Section 2.1.a.

Logistic Support Analysis record (LSAR) - A document that provides detailed data generated by an LSA process concerning logistic support requirements. For details, see Section 2.1.b.

Integrated Weapon System Data Base (IWSDB) - Both the conceptual definition of and the total logical collection itself of shared product definition data and logistic support data for a specific weapon system. For details, see Section 2.1.e.

Metadata - Data that describes other data.

Nijssen Information Analysis Method (NIAM) - An information modeling methodology based on the binary relationship technique. For details, see Sections 2.2.m and 2.2.p.

Normalization - A process of obtaining "stable" groupings of attributes and relations.

Output Specification (OS) - An SGML construct that describes the formatting and style options for a class of documents whose structure is defined by a particular Document Type Definition (DTD). For details, see Section 2.1.d.

Product definition Data (PDD) - The data that comprises the information needed for the design, analysis, manufacture, testing, and inspection of a product. For details, see Section 2.1.e.

Product Data Exchange Standard (PDES) - A voluntary standardization effort to provide a complete, unambiguous, digital data definition of the physical and functional characteristics of each element/part of a product throughout its life cycle.

Schema - A diagram describing the semantic or relational structure of data.

Standard Generalized Markup language (SGML) - A methodology used to define the structure of documents in terms of an abstract and unambiguous notation, independent of application and input/output devices. Five relevant SGML concepts are: the SGML declaration, the Document Type Definition (DTD), the Document Instance (DI), the Output Specification (OS), and Formatted Output Specification Instance (FOSI). For details, see Section 2.2.h.

SQL - The standard data definition and data manipulation language for a relational database.

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix C - NIAM Modeling Guide.

30 NIAM Modeling Guide.

NIAM or binary semantic modeling attempts to achieve a binary representation of the relationships between object classes. NIAM emphasizes presentation the meaning of information rather than its structure. When any meaningful relationship involves more than two object classes, population tables are constructed to ensure the uniqueness and the correctness of the relationship.

NIAM conceptual schema diagrams can be represented graphically by objects, roles of objects, and constraints on roles and objects. The following is a definition of the symbols which are used in the NIAM binary relationship modeling of Appendix E and which were primarily derived from Sections 2.2.m and 2.2.a.

30.1 Objects - Objects are real or abstract entities. Two classes of objects are used in NIAM, they are entity types and label types.



Entity Type - A collection of objects that have similar properties. An entity type is represented graphically as a solid circle and referenced by "label type" in database.

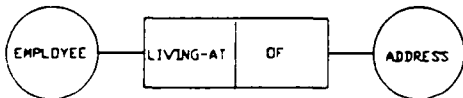
Example: "Employee" is the set of all people who are hired.



Label Type - A label type is used to reference a particular entity type. A label type is represented graphically as broken circle. Label types are lexical objects in database tables that are used to refer to entity types. Example: "surname"

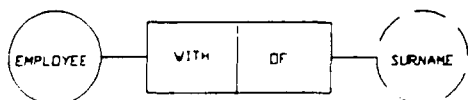
30.2 Roles - Roles indicate relationships between objects.

The roles played by entity types are explicitly shown as boxes. A line joining a role to an entity type indicates that the role is played by entities of that type. The name of the role is placed inside or beside the box.



Fact Type - A relationship between two entity types.

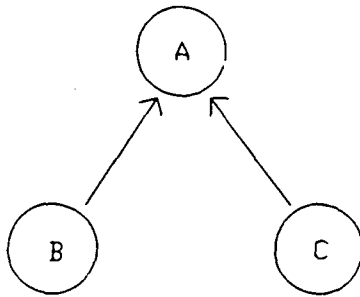
Example: The role "living at" describes the relationship between the entity types of "employee" and "address".



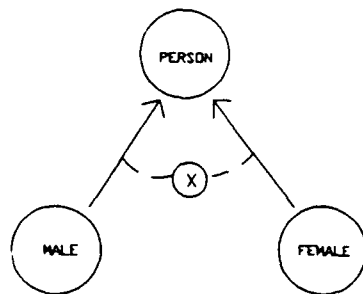
Reference Type - A relationship between a entity type and a label type. It provides a bridge between the objects and the labels. Example: The label type "surname" is used to identify an entity type, such as "employee".

30.3 Constraints on Objects and Roles - This Section lists various constraints or restrictions on objects and roles. The following conventions are used: a member or members of entity type A is designated as "A", a member or members of entity type B is designated as "B", and a member or members of entity type C is designated as "C".

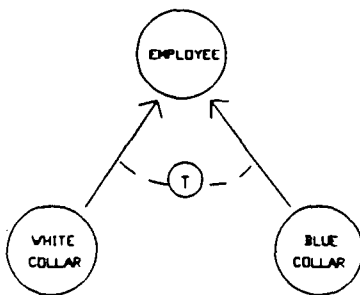
a. Subtype Constraint - Subtype constraints are the rules which restrict the division of entity types into subtypes (subsets). A subtype shares (inherits) all properties of its supertype. A subtype is represented graphically by a single head arrow.



Subtype - Entity types B and C are subtypes of entity type A.
Entity type A is a common supertype of entity types B and C.
Entity type A can have zero, one or many subtypes.



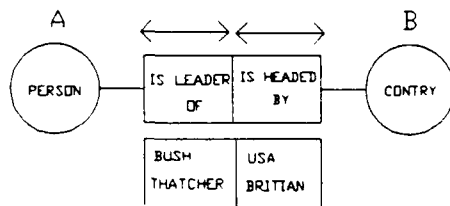
Exclusion subtype - Entity types B and C are disjoint subsets of entity type A.
Example: Subtypes "male" and "female" are disjoint subsets of supertype "person".



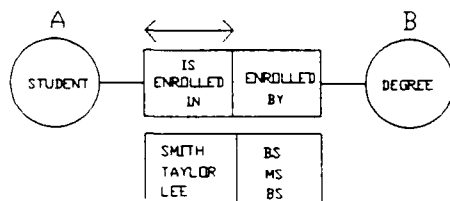
Total subtype - All of the members of the supertype A are contained in the subtypes B and C. The subtypes B and C may intersect.

Example: All employees are either white collar or blue collar (or both).

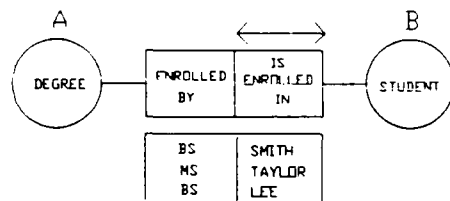
b. Uniqueness Constraints - The uniqueness constraint is represented by placing a double headed arrow above or below the role box. Marking a single role with a uniqueness bar means that no duplicates of entries in the associated column of the population table are allowed. A uniqueness bar that spans a whole row means that each entry in the row is unique to the population table.



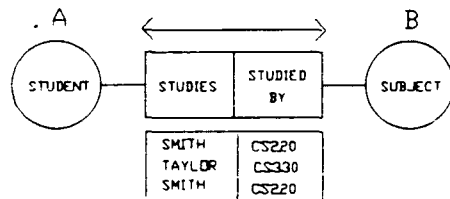
One to One - Each "A" relates to at most one "B". Each "B" relates to at most one "A". No duplications are allowed in either columns of the population table.
Example: A person may be leader of only one country. One country is headed by one leader.



Many to One - Each "A" relates to at most one "B". No duplications are allowed in left column.
Example: Many students can be enrolled in the one degree program, but a student may enroll in at most one degree program.

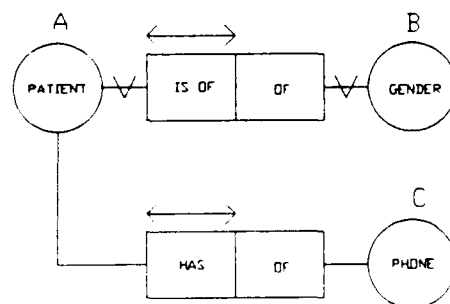


One to Many - Each "A" may relate to many "B". No duplications are allowed in right column.
Example: A degree program can be enrolled by many students.



Many to Many - Each "A" may relate to many "B". Each "B" may relate to many "A". No duplications are allowed in combination of both columns.
Example: A student studies many subjects. A subject can be studied by many students. But a student cannot study the same subject twice.

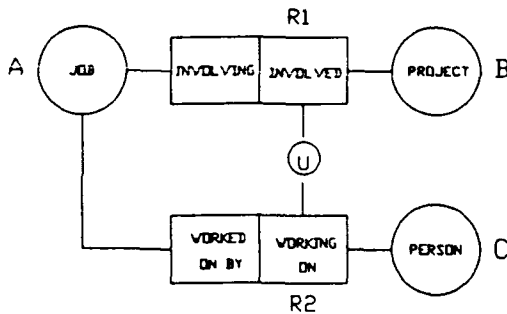
c. Mandatory/Optional Constraints - A mandatory constraint specifies whether or not certain information must be recorded. It is indicated by placing a "V" on the line connecting the role to the entity type.



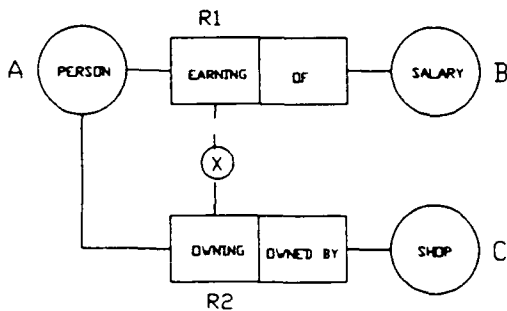
Mandatory - Each "A" relates to one and only one "B". Each "B" relates to one or many "A".
Example: The gender of every patient must be recorded and there is a patient of every gender.

Optional - Each "A" may relate to at most one "C". Each "C" may relate to many "A".
Example: A patient may or may not have a telephone.

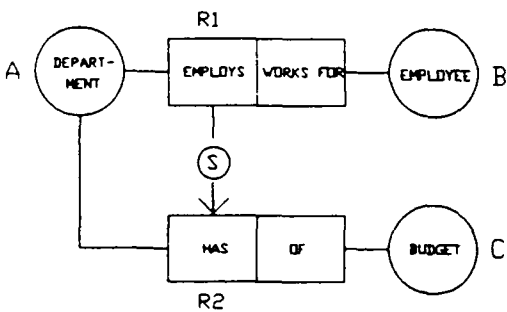
d. Multiple Role Constraints - Multiple role constraints are shown as a line between the affected roles with a circle containing the constraint letter.



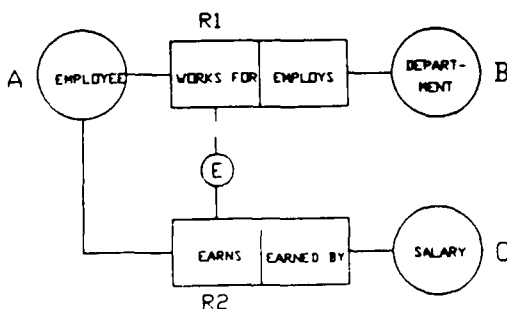
Joint uniqueness constraints - "B" playing role R1 with "A" and "C" playing role R2 with "A" are both required to uniquely identify "A". Example: Both "person" and "project" are required to complete the "job".



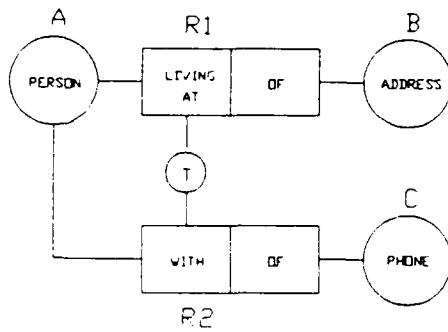
Exclusion constraints - Any "A" that is related to "B" through role R1 cannot be related to "C" through Role R2 and vice versa. Example: A person can either earn a salary or own a shop, but not both.



Subset constraints - "A" which play role R2 with "C" are a subset of "A" which play role R1 with "B". Example: A department can employ person only if that department has a budget, or the set of departments employing person is a subset of departments having budgets.

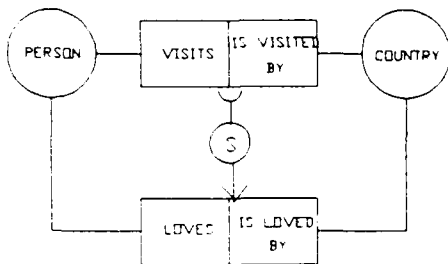


Equality constraints - All "A" that are related to "B" through role R1 also are related to "C" through role R2 and vice versa. Example: Every employee works for a department earns a salary and every employee who earns a salary must also work for a department.



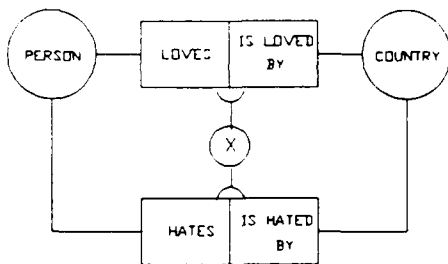
Joint total constraints - "A" may play role R1 with "B". "A" may also play role R2 with "C". Example: A person live at the address has the same telephone number. (Inclusive OR)

e. Role Pair Constraints



Subset constraints -

Example: For any person the set of countries he visits is a subset of the set of countries he loves and for any countries the set of people who visit it is a subset of the people who love it.



Exclusive constraints -

Example: For any country the set of people who love it is disjoint from the set of people who hate it, and for any person, the set of countries he loves is disjoint from the set of countries he hates.

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix D - CDM Specification.

40 CDM Specification.

<!DOCTYPE techinfo [

<!-- AF Content Data Model (AFCDM) - Version 5.1 8/23/89, with
modifications by AFCDM Review Board Meeting on 11/2/89.

This document is an SGML Document Type Definition (DTD) which describes the logical structure for a database of technical information for AFCDM. This version is a working draft which is still under development by the Air Force Human Resources Laboratory (AFHRL/LRC) at Wright Patterson, AFB. For details, see Section 2.1.g - "Draft Specification for Digital Technical Information".

<!-- Entity Declarations - These entity declarations define abbreviations for a set of frequently used attributes: "id", "refid".

<!ENTITY % ids "id ID #REQUIRED
refid NMTOKEN #REQUIRED">

<!ENTITY % refid "NMTOKEN">

<!ENTITY % refids "NMTOKENS">

<!-- 40.1 CDM-ELEMENT -->

There are no corresponding SGML/DTD statements (See Section 50.1.1).

<!-- 40.2 COMMON-ELEMENT -->

There are no corresponding SGML/DTD statements (See Section 50.1.2).

<!-- 40.3 TECHINFO, SYSTEM, OPERINFO and DESCINFO -->

<!ELEMENT techinfo - - (system*, operinfo*, descinfo*,
task*, step*, reqcond*, person*, equip*, comsum*,
verb*, context*, warning*, caution*, note*,
annot*, partinfo*, partbase*, xref*, faultinf*,
test*, outcome*, fltstate*, fault*, rect*,
text*, dictitem*, table*, colhddef*, entry*,
graphic*, grphprim*, video*, audio*, process*,
prompt*, fillin*, menu*, choice*, assertion*,
precond*, property*, value*)>

<!ATTLIST techinfo %ids;
system %refids; #REQUIRED
name CDATA #IMPLIED
type CDATA #IMPLIED
itemid CDATA #IMPLIED
context %refids; #IMPLIED
xref IDREFS #IMPLIED>

<!ELEMENT	system	- o	EMPTY >
<!ATTLIST	system	%ids;	
		name	CDATA #IMPLIED
		type	(vehicle /system /subsystem) #IMPLIED
		itemid	CDATA #IMPLIED
		context	%refids; #IMPLIED
		xref	IDREFS #IMPLIED
		system	%refids; #IMPLIED
		operinfo	%refids; #IMPLIED
		descinfo	%refids; #IMPLIED
		task	%refids; #IMPLIED
		partinfo	%refids; #IMPLIED
		faultinf	%refids; #IMPLIED>

<!ELEMENT	operinfo	- o	EMPTY >
<!ATTLIST	operinfo	%ids;	
		name	CDATA #IMPLIED
		type	CDATA #IMPLIED
		itemid	CDATA #IMPLIED
		context	%refids; #IMPLIED
		xref	IDREFS #IMPLIED
		descinfo	%refids; #IMPLIED
		task	%refids; #IMPLIED>

<!ELEMENT	descinfo	- o	EMPTY >
<!ATTLIST	descinfo	%ids;	
		name	CDATA #IMPLIED
		type	CDATA #IMPLIED
		itemid	CDATA #IMPLIED
		context	%refids; #IMPLIED
		xref	IDREFS #IMPLIED
		assertion	IDREFS #IMPLIED
		descinfo	%refids; #IMPLIED
		text	%refid; #IMPLIED
		table	%refids; #IMPLIED
		graphic	%refids; #IMPLIED
		audio	%refids; #IMPLIED
		video	%refids; #IMPLIED
		process	%refids; #IMPLIED
		annot	%refids; #IMPLIED
		prompt	%refids; #IMPLIED>

<!-- 40.4 TASK, REQCOND, PERSON, WARNING, CAUTION, NOTE, and VERB -->

<!ELEMENT	task	- o	EMPTY >
<!ATTLIST	task	%ids;	
		step	%refids; #REQUIRED
		name	CDATA #IMPLIED
		type	CDATA #IMPLIED
		itemid	CDATA #IMPLIED
		context	%refids; #IMPLIED

		xref	IDREFS	#IMPLIED
		esttime	NUTOKEN	#IMPLIED
		verb	%refids;	#IMPLIED
		reqcond	%refids;	#IMPLIED
		person	%refids;	#IMPLIED
		equip	%refids;	#IMPLIED
		consum	%refids;	#IMPLIED
		warning	%refids;	#IMPLIED
		caution	%refids;	#IMPLIED
		note	%refids;	#IMPLIED
		followon	%refids;	#IMPLIED>
<!ELEMENT	reqcond	- o	EMPTY >	
<!ATTLIST	reqcond	%ids;		
		name	CDATA	#IMPLIED
		type	CDATA	#IMPLIED
		itemid	CDATA	#IMPLIED
		context	%refids;	#IMPLIED
		xref	IDREFS	#IMPLIED
		precond	IDREFS	#IMPLIED
		elmntref	%refids;	#IMPLIED>
<!ELEMENT	person	- o	EMPTY >	
<!ATTLIST	person	%ids;		
		qty	CDATA	#REQUIRED
		name	CDATA	#IMPLIED
		type	CDATA	#IMPLIED
		itemid	CDATA	#IMPLIED
		context	%refids;	#IMPLIED
		xref	IDREFS	#IMPLIED>
<!ELEMENT	warning	- o	EMPTY >	
<!ATTLIST	warning	%ids;		
		text	%refid;	#REQUIRED
		name	CDATA	#IMPLIED
		type	CDATA	#IMPLIED
		itemid	CDATA	#IMPLIED
		context	%refids;	#IMPLIED
		xref	IDREFS	#IMPLIED>
<!ELEMENT	caution	- o	EMPTY >	
<!ATTLIST	caution	%ids;		
		text	%refid;	#REQUIRED
		name	CDATA	#IMPLIED
		type	CDATA	#IMPLIED
		itemid	CDATA	#IMPLIED
		context	%refids;	#IMPLIED
		xref	IDREFS	#IMPLIED>
<!ELEMENT	note	- o	EMPTY >	
<!ATTLIST	note	%ids;		
		text	%refid;	#REQUIRED

		name	CDATA	#IMPLIED
		type	CDATA	#IMPLIED
		itemid	CDATA	#IMPLIED
		context	%refids;	#IMPLIED
		xref	IDREFS	#IMPLIED>
<!ELEMENT	verb	- o	EMPTY >	
<!ATTLIST	verb	%ids;		
		name	CDATA	#IMPLIED
		type	CDATA	#IMPLIED
		itemid	CDATA	#IMPLIED
		context	%refids;	#IMPLIED
		xref	IDREFS	#IMPLIED>
 <!-- 40.5 STEP, EQUIP and ANNOT -->				
<!ELEMENT	step	- o	EMPTY >	
<!ATTLIST	step	%ids;		
		text	%refid;	#REQUIRED
		name	CDATA	#IMPLIED
		type	CDATA	#IMPLIED
		itemid	CDATA	#IMPLIED
		context	%refids;	#IMPLIED
		xref	IDREFS	#IMPLIED
		esttime	NUTOKEN	#IMPLIED
		assertion	IDREFS	#IMPLIED
		step	%refids;	#IMPLIED
		table	%refids;	#IMPLIED
		graphic	%refids;	#IMPLIED
		audio	%refids;	#IMPLIED
		video	%refids;	#IMPLIED
		process	%refids;	#IMPLIED
		annot	%refids;	#IMPLIED
		prompt	%refids;	#IMPLIED
		warning	%refids;	#IMPLIED
		caution	%refids;	#IMPLIED
		note	%refids;	#IMPLIED
		verb	%refids;	#IMPLIED
		reqcond	%refids;	#IMPLIED
		person	%refids;	#IMPLIED
		equip	%refids;	#IMPLIED
		consum	%refids;	#IMPLIED>
<!ELEMENT	equip	- o	EMPTY >	
<!ATTLIST	equip	%ids;		
		qty	CDATA	#REQUIRED
		equip	%refids	#IMPLIED
		name	CDATA	#IMPLIED
		type	CDATA	#IMPLIED
		itemid	CDATA	#IMPLIED
		context	%refids;	#IMPLIED
		xref	IDREFS	#IMPLIED>

<!ELEMENT	annot	- o	EMPTY >	
<!ATTLIST	annot	%ids;		
		text	%refid;	#REQUIRED
		name	CDATA	#IMPLIED
		type	CDATA	#IMPLIED
		itemid	CDATA	#IMPLIED
		context	%refids;	#IMPLIED
		xref	IDREFS	#IMPLIED
		user	CDATA	#IMPLIED>

<!-- 40.6 PROMPT, FILLIN, MENU, and CHOICE -->

<!ELEMENT	prompt	- o	EMPTY >	
<!ATTLIST	prompt	%ids;		
		name	CDATA	#IMPLIED
		type	CDATA	#IMPLIED
		itemid	CDATA	#IMPLIED
		context	%refids;	#IMPLIED
		xref	IDREFS	#IMPLIED
		text	%refid;	#IMPLIED
		fillin	%refids;	#IMPLIED
		menu	%refids;	#IMPLIED>

<!ELEMENT	fillin	- o	EMPTY >	
<!ATTLIST	fillin	%ids;		
		text	%refid;	#REQUIRED
		property	IDREF	#REQUIRED
		name	CDATA	#IMPLIED
		type	CDATA	#IMPLIED
		itemid	CDATA	#IMPLIED
		context	%refids;	#IMPLIED
		xref	IDREFS	#IMPLIED
		range	CDATA	#IMPLIED
		default	CDATA	#IMPLIED>

<!ELEMENT	menu	- o	EMPTY >	
<!ATTLIST	menu	%ids;		
		text	%refid;	#REQUIRED
		property	IDREF	#REQUIRED
		choice	IDREFS	#REQUIRED
		name	CDATA	#IMPLIED
		type	CDATA	#IMPLIED
		itemid	CDATA	#IMPLIED
		context	%refids;	#IMPLIED
		xref	IDREFS	#IMPLIED
		select	(single multiple)	"single"
		default	IDREFS	#IMPLIED>

<!ELEMENT	choice	- o	EMPTY >	
<!ATTLIST	choice	id	ID	#REQUIRED
		text	%refid;	#REQUIRED

		value	IDREFS	#REQUIRED>
<!-- 40.7 PRECOND -->				
<!ELEMENT	precond	- o	EMPTY >	
<!ATTLIST	precond	id	ID	#REQUIRED
		property	IDREF	#REQUIRED
		value	IDREFS	#REQUIRED
		polarity	(pos neg)	"pos"
		op	(eq lt lte gt gte in)	"eq">
<!-- 40.8 ASSERTION, PROPERTY and VALUE -->				
<!ELEMENT	assertion	- o	EMPTY>	
<!ATTLIST	assertion	id	ID	#REQUIRED
		property	IDREF	#REQUIRED
		value	IDREFS	#REQUIRED>
<!ELEMENT	property	- o	EMPTY>	
<!ATTLIST	property	id	ID	#REQUIRED
		text	%refid;	#REQUIRED
		elmntref	%refid;	#IMPLIED>
<!ELEMENT	value	- o	EMPTY>	
<!ATTLIST	value	id	ID	#REQUIRED
		text	%refid;	#REQUIRED>
<!-- 40.9 AUDIO, VIDEO and PROCESS -->				
<!ELEMENT	audio	- o	EMPTY >	
<!ATTLIST	audio	%ids;		
		name	CDATA	#IMPLIED
		type	CDATA	#IMPLIED
		itemid	CDATA	#IMPLIED
		context	%refids;	#IMPLIED
		xref	IDREFS	#IMPLIED
		file	CDATA	#IMPLIED
		exrefid	CDATA	#IMPLIED>
<!ELEMENT	video	- o	EMPTY >	
<!ATTLIST	video	%ids;		
		name	CDATA	#IMPLIED
		type	CDATA	#IMPLIED
		itemid	CDATA	#IMPLIED
		context	%refids;	#IMPLIED
		xref	IDREFS	#IMPLIED
		file	CDATA	#IMPLIED
		exrefid	CDATA	#IMPLIED>
<!ELEMENT	process	- o	EMPTY >	
<!ATTLIST	process	%ids;		

name	CDATA	#IMPLIED
type	CDATA	#IMPLIED
itemid	CDATA	#IMPLIED
context	%refids;	#IMPLIED
xref	IDREFS	#IMPLIED
file	CDATA	#IMPLIED
exrefid	CDATA	#IMPLIED>

<!-- 40.10 TEXT and DICTITEM -->

<!ELEMENT	text	- -	(#PCDATA attvalue)+ >
<!ATTLIST	text	%ids;	
	name	CDATA	#IMPLIED
	type	CDATA	#IMPLIED
	itemid	CDATA	#IMPLIED
	context	%refids;	#IMPLIED
	xref	IDREFS	#IMPLIED>

<!ELEMENT	dictitem	- -	EMPTY >
<!ATTLIST	dictitem	%ids;	
	name	CDATA	#REQUIRED
	elmntref	%refids	#REQUIRED
	type	(gloss abbsym symbol other)	"other"
	itemid	CDATA	#IMPLIED
	context	%refids;	#IMPLIED
	xref	IDREFS	#IMPLIED>

<!ELEMENT	attvalue	- o	EMPTY >
<!ATTLIST	attvalue	elmntref	%refid;
	attname	NAME	"name">

<!-- 40.11 CONSUM -->

<!ELEMENT	consum	- o	EMPTY >
<!ATTLIST	consum	%ids;	
	milspec	CDATA	#REQUIRED
	mfgcode	CDATA	#REQUIRED
	govstd	CDATA	#REQUIRED
	qty	CDATA	#REQUIRED
	name	CDATA	#IMPLIED
	type	CDATA	#IMPLIED
	itemid	CDATA	#IMPLIED
	context	%refids;	#IMPLIED
	xref	IDREFS	#IMPLIED
	uom	CDATA	#IMPLIED>

<!-- 40.12 TABLE, COLHDDEF, and ENTRY -->

<!ELEMENT	table	- -	EMPTY >
<!ATTLIST	table	%ids;	
	colhddef	IDREFS	#REQUIRED

		entry	IDREFS	#REQUIRED
		name	CDATA	#IMPLIED
		type	CDATA	#IMPLIED
		itemid	CDATA	#IMPLIED
		context	%refids;	#IMPLIED
		xref	IDREFS	#IMPLIED>
<!ELEMENT	colhddef	- o	EMPTY>	
<!ATTLIST	colhddef	id	ID	#REQUIRED
		colnum	NUTOKEN	#REQUIRED
		name	CDATA	#IMPLIED
		type	CDATA	#IMPLIED>
<!ELEMENT	entry	- o	EMPTY>	
<!ATTLIST	entry	id	ID	#REQUIRED
		col	NUTOKEN	#REQUIRED
		row	NUTOKEN	#REQUIRED
		text	%refid;	#IMPLIED
		elmntref	%refid;	#IMPLIED>
<!-- 40.13 GRAPHIC -->				
<!ELEMENT	graphic	- o	EMPTY >	
<!ATTLIST	graphic	%ids;		
		name	CDATA	#IMPLIED
		type	(normal locat overlay schem functblk wiring engin icons)	
		itemid	CDATA	#IMPLIED
		context	%refids;	#IMPLIED
		xref	IDREFS	#IMPLIED
		text	%refid;	#IMPLIED
		focus	%refids;	#IMPLIED
		transfrm	NUTOKENS	#IMPLIED
		window	NUTOKENS	#IMPLIED
		penshape	CDATA	#IMPLIED
		penpatt	CDATA	#IMPLIED
		minsize	NUTOKENS	#IMPLIED
		grphprim	%refids	#IMPLIED>
<!-- 40.14 GRPHPRIM -->				
<!ELEMENT	grphprim	- -	#PCDATA >	
<!ATTLIST	grphprim	%ids;		
		name	CDATA	#IMPLIED
		type	(normal locat overlay schem functblk wiring engin)	#IMPLIED
		itemid	CDATA	#IMPLIED
		context	%refids;	#IMPLIED
		xref	IDREFS	#IMPLIED
		text	%refid;	#IMPLIED

file	CDATA	#IMPLIED
coding	(cgmlchar cgmlbin cgmlclear fax iges dxf gks) "cgmlbin"	
transfrm	NUTOKENS	#IMPLIED
window	NUTOKENS	#IMPLIED
penshape	CDATA	#IMPLIED
penpatt	CDATA	#IMPLIED
minsize	NUTOKENS	#IMPLIED
start	CDATA	#IMPLIED
stop	CDATA	#IMPLIED

<!-- 40.15 FAULTINFO, TEST and OUTCOME -->

<!ELEMENT	faultinf	- o	EMPTY >
<!ATTLIST	faultinf	%ids;	
	test	%refids;	#REQUIRED
	fault	%refids;	#REQUIRED
	name	CDATA	#IMPLIED
	type	CDATA	#IMPLIED
	itemid	CDATA	#IMPLIED
	context	%refids;	#IMPLIED
	xref	IDREFS	#IMPLIED

<!ELEMENT	test	- o	EMPTY >
<!ATTLIST	test	%ids;	
	task	%refids;	#REQUIRED
	outcome	%refids;	#REQUIRED
	name	CDATA	#IMPLIED
	type	CDATA	#IMPLIED
	itemid	CDATA	#IMPLIED
	context	%refids;	#IMPLIED
	xref	IDREFS	#IMPLIED
	text	%refid;	#IMPLIED
	agent	(human machine)	"human"
	range	CDATA	#IMPLIED

<!ELEMENT	outcome	- o	EMPTY >
<!ATTLIST	outcome	%ids;	
	precond	IDREFS	#REQUIRED
	fltstate	%refids;	#REQUIRED
	name	CDATA	#IMPLIED
	type	CDATA	#IMPLIED
	itemid	CDATA	#IMPLIED
	context	%refids;	#IMPLIED
	xref	IDREFS	#IMPLIED
	text	%refid;	#IMPLIED

<!-- 40.16 FAULT, FLTSTATE and RECT -->

<!ELEMENT	fault	- o	EMPTY >
<!ATTLIST	fault	%ids;	
	rect	%refids;	#REQUIRED

		name	CDATA	#IMPLIED	
		type	CDATA	#IMPLIED	
		itemid	CDATA	#IMPLIED	
		context	%refids;	#IMPLIED	
		xref	IDREFS	#IMPLIED	
		mtbf	CDATA	#IMPLIED	
		fltstate	%refids;	#IMPLIED	
		text	%refid;	#IMPLIED	
		partinfo	%refids;	#IMPLIED>	
<!ELEMENT	fltstate	- o	EMPTY >		
<!ATTLIST	fltstate	%ids;			
		name	CDATA	#IMPLIED	
		type	CDATA	#IMPLIED	
		itemid	CDATA	#IMPLIED	
		context	%refids;	#IMPLIED	
		xref	IDREFS	#IMPLIED	
		text	%refid;	#IMPLIED	
		expfault	%refids;	#IMPLIED	
		impfault	%refids;	#IMPLIED	
		weight	NUTOKENS	#IMPLIED	
		test	%refid;	#IMPLIED>	
<!ELEMENT	rect	- o	EMPTY >		
<!ATTLIST	rect	%ids;			
		task	%refids;	#REQUIRED	
		name	CDATA	#IMPLIED	
		type	CDATA	#IMPLIED	
		itemid	CDATA	#IMPLIED	
		context	%refids;	#IMPLIED	
		xref	IDREFS	#IMPLIED	
		action	(swap maint)	"swap"	
		agent	(human machine)	"human"	
		text	%refid;	#IMPLIED	
		test	%refids;	#IMPLIED	
		fault	%refids;	#IMPLIED>	
<!-- 40.17	PARTINFO	-->			
<!ELEMENT	partinfo	- o	EMPTY >		
<!ATTLIST	partinfo	%ids;			
		partbase	IDREFS	#REQUIRED	
		refdes	NMTOKEN	#REQUIRED	
		unitsper	NUTOKEN	#REQUIRED	
		indxnum	NUTOKEN	#REQUIRED	
		usablon	NUTOKEN	#REQUIRED	
		mtbf	CDATA	#REQUIRED	
		graphic	%refids;	#REQUIRED	
		name	CDATA	#IMPLIED	
		type	CDATA	#IMPLIED	
		itemid	CDATA	#IMPLIED	
		context	%refids;	#IMPLIED	

		xref	IDREFS	#IMPLIED
		nounid	NUTOKEN	#IMPLIED
		nountype	NUTOKEN	#IMPLIED
		replvl	CDATA	#IMPLIED>

<!-- 40.18 PARTBASE -->

<!ELEMENT	partbase	- o	EMPTY >	
<!ATTLIST	partbase	%ids;		
		partnum	CDATA	#REQUIRED
		cage	CDATA	#REQUIRED
		nsn	CDATA	#REQUIRED
		smr	CDATA	#REQUIRED
		hci	CDATA	#REQUIRED
		name	CDATA	#IMPLIED
		type	CDATA	#IMPLIED
		itemid	CDATA	#IMPLIED
		xref	IDREFS	#IMPLIED>

<!-- 40.19 CONTEXT -->

<!ELEMENT	context	- o	EMPTY >	
<!ATTLIST	context	id	ID	#REQUIRED
		security	(uc c s ts)	#IMPLIED
		release	NMTOKENS	#IMPLIED
		config	NMTOKENS	#IMPLIED
		maintlvl	CDATA	#IMPLIED
		track	NUTOKENS	#IMPLIED
		version	NUTOKENS	#IMPLIED
		valstat	CDATA	#IMPLIED
		verstat	CDATA	#IMPLIED
		precond	IDREFS	#IMPLIED>

<!-- 40.20 XREF and LIST -->

<!ELEMENT	xref	- o	EMPTY >	
<!ATTLIST	xref	id	ID	#REQUIRED
		relation	CDATA	#IMPLIED
		elmntref	%refids;	#IMPLIED
		attname	NAMES	#IMPLIED
		exrefids	CDATA	#IMPLIED>

<!ELEMENT	list	- o	EMPTY >	
<!ATTLIST	list	%ids;		
		elmntref	%refids;	#REQUIRED
		name	CDATA	#IMPLIED
		context	%refids;	#IMPLIED>

]>

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix E - CDM/NIAM Conceptual Schema

50.1 CDM/NAIM Conceptual Schema Descriptions

The following are descriptions of the CDM Conceptual Diagrams contained in Section 50.2. This Section is derived from Section 40. The NIAM, Appendix C, is used as the modeling language.

50.1.1 CDM-ELEMENT (See Diagram (or Section) 50.2.1)

The CDM-ELEMENT is the top level data element of the CDM information model structure. All the elements in the CDM have the unique label type identifier "id". All the twelve subtype elements shown in this Diagram inherit this label type "id".

50.1.2 COMMON-ELEMENT (See Diagram 50.2.2)

All the thirty three subtype elements in this Diagram inherit these six common entity or label types ("refid", "name", "itemid", "type", "context" and "xref") in addition to the label type "id" which is inherited from COMMON-ELEMENT in Diagram 50.1.1.

The entity "refid" (reference identifier) is used by data elements to refer or "point" to other elements. Since "refid" is not unique, it may refer to several elements in the database. These referenced elements will have different "context" such as version number or security level (see Diagram 50.2.20) which will be used to determine which unique element is appropriate for a particular situation.

Most elements have a set of attributes ("name", "type", "itemid", and "xref") which describe the nature of the element's content. These include attributes defining the element's "name" and "type". There is also an "itemid" attribute used to indicate which piece of equipment (or "item") is related to the information element. The "itemid" could be a reference designator, a SSN number, a part number, etc., depending on the item of interest. There is also an attribute "xref" which defines relational links between the element and other elements in the database. An "xref" specifies the reference identifier for a related element and the type of relation being specified.

50.1.3 TECHINFO, SYSTEM, OPERINFO and DESCINFO (See Diagram 50.2.3)

The "techinfo" element is the top element of the CDM technical information. Its content is a long list of data elements which comprise the raw data "records" or "tables" of the CDM database.

The "system" element of the database is the root, or top level in the system/end-item hierarchy.

The CDM specifies a hierarchically organized database of technical information for a weapon system. The main hierarchy of the database parallels the equipment hierarchy of the weapon system. This is normally

represented as a vehicle - system - subsystem - subassembly hierarchy in Technical Orders. That hierarchy is represented in the CDM by the "system" element specified. Here "system" is used in its most generic sense, meaning any component or item in the equipment hierarchy. A "system" could represent the vehicle, an aircraft system, subsystem, or subassembly.

At any level in this hierarchy, elements may reference associated information. They may reference procedural task information ("task"), descriptive information ("descinfo"), parts information ("partinfo"), fault information ("faultinf"), or operational information ("operinfo"). This information should be attached to the level where it is most appropriate. For example, vehicle towing procedures should be referenced at the "vehicle" level. The removal task for a circuit card in the radar should be referenced at the radar "system" level.

The element "descinfo" is used to define general purpose, non-procedural, narrative information such as theory of operation, schematics, wiring diagrams, etc. "descinfo" is a very flexible, general purpose information node. It can be used to describe any arbitrary, hierarchical hypertext-like node containing sub-paragraphs ("descinfo"), data ("text", "table", "graphic", "annot", "audio", "video", or "process"), user interaction instructions ("prompt"), and assertion properties ("assertion") which are asserted whenever "descinfo" is read.

50.1.4 TASK, REQCOND, PERSON, WARNING, CAUTION, NOTE and VERB (See Diagram 50.2.4)

The elements "task", and "step" define a maintenance task or procedure. A "task" consists of a list of steps ("step"), a list of followon tasks ("followon"), and the attributes estimated time ("esttime") and action describing "verb". The elements "reqcond" (required conditions), "person" (personnel required), "equip" (equipment required), and "consum" (consumables). The additional elements "warning", "caution", "note", and "annot" (user annotations) are referenced by "step".

50.1.5 STEP, EQUIP and ANNOT (See Diagram 50.2.5)

See above Section 50.1.4 above.

50.1.6 PROMPT, FILLIN, MENU and CHOICE (See Diagram 50.2.6)

A "prompt" specifies either a fill-in-the-blank ("fillin") or menu choice ("menu") question for the user. Prompts are characterized in terms of property-value pairs similar to assertions and preconditions). Basically, each prompt is associated with a "property" which specifies the property which will be asserted along with the user's response when the prompt is answered. If the prompt is a "fillin" the user's response will be asserted as the "value" of the specified "property". If the prompt is a "menu", the user's "choice" selection from the menu will have an associated "value" which will be asserted as the "value" of the prompt's "property". Once this assertion is made, other elements

in the system may use the information to test preconditions ("precond") concerning the asserted property.

The "text" of a prompt is the question which will be displayed to the user. The "text" of a "choice" is the list of possible menu selections which will be displayed to the user.

Both "fillin" and "menu" prompts can have a "default" value. In the case of a "fillin", the "default" is a text string which will be used as the initial entry in the fill-in-the-blank form. In the case of a "menu", the default will be an "address" of the possible "choice" responses.

50.1.7 PRECOND (See Diagram 50.2.7)

"precond" and "assertion" are both defined in terms of property-value pairs. A "property" is any "text" string which defines a property. A "value" is another "text" string which defines a value. Property-pairs may be asserted or tested by the run time presentation software. An "assertion" on a step will be asserted whenever that step is performed. A "precond" is a test of a property previously asserted. The property element also has an optional "elmtref" attribute used to indicate a prompt or task which can be activated to acquire a value for the property if none has been asserted.

50.1.8 ASSERTION, PROPERTY and VALUE (See Diagram 50.2.8)

See above Section 50.1.7 above.

50.1.9 AUDIO, VIDEO and PROCESS (See Diagram 50.2.9)

The elements "audio", "video", and "process" are references to either a file name or an external source which contains an audio sequence, a video sequence, or a software process, respectively.

50.1.10 TEXT and DICTITEM (See Diagram 50.2.10)

"text" is the primitive text element referenced by more complex data elements of the CDM. A "text" unit is basically a text string of "parsable character data" (#PCDATA). Within a text string, attribute values ("attvalue") of other CDM elements may be referenced and inserted as text strings. For example, the string may contain a reference to a standard system name, or a standard part nomenclature, or a standard task name. "attvalue" may be used to embed one of these references in a string which tells the display system to find the value of the referenced attribute and place that value into the text string for display. By using this mechanism, standard terminology can be referenced consistently throughout the database, and any changes to the standard terminology can be made in one location and automatically updated throughout the database.

50.1.11 COMSUM (See Diagram 50.2.11)

The "consum" element defines the consumables required in the maintenance "step" procedures.

50.1.12 TABLE, COLHDDEF and ENTRY (See Diagram 50.2.12)

The elements "table", "colhddef", and "entry" define the structure for a table of information. The cells or "entries" of a table may be a "text" unit or any element identified by an "refid".

50.1.13 GRAPHIC (See Diagram 50.2.13)

The CDM allows graphics to be referenced from external graphics files or embedded graphics files in the CDM database. The element "grphprim" may contain a "file" name which identifies an external file containing a graphic data in any of the enumerated formats (cgm, iges, dxf, fax, etc.). The same graphic data may also be included directly in the CDM by putting the data in the "#PCDATA" content portion of a "grphprim" element.

Both "graphic" and "grphprim" have a set of optional attributes to specify transformations (i.e., scaling, translating, rotating, clipping, etc.). A "graphic" or "grphprim" element may specify a transformation matrix ("transform"), a clipping ("window"), a pen shape ("penshape"), a pen pattern ("penpatt"), and a label ("text"). Transformations ("transfrm") are specified by a 9-number transformation matrix which specifies coordinate translation, scaling, reflection, and rotation in terms of homogeneous coordinates.

Composite graphics may be constructed by grouping transformed graphics ("graphic" or "grphprim") into "graphic" elements. These transformed, labeled, named, and typed graphic illustrations are then referenced by steps in the CDM. A composite "graphic" may also specify a list of "focus" objects which are the subgraphics of interest for that particular illustration. This attribute could be used to specify which subgraphics in an illustration are to be highlighted, labeled, etc., by the presentation software.

"graphic" and "grphprim" also may specify a minimum size ("minsize") required to satisfactorily display the graphic to the user. The minimum size is specified in terms of the visual angle the graphic image should subtend on the viewer's eye. This will allow different display systems with different viewing distances to adjust the physical size of the graphic to provide the correct visual image intended by the author.

50.1.14 GRPHPRIM (See Diagram 50.2.14)

See above Section 50.1.13 above.

50.1.15 FAULTINFO, TEST, and OUTCOME (See Diagram 50.2.15)

Three types of fault information can be described in the CDM:
(1) fault reporting decision trees, (2) fault isolation decision trees,

and (3) dynamic fault isolation models (such as AFHRL's MDAS model). The fault reporting and isolation decision trees are static, predefined decision sequences. A dynamic fault model generates the decision sequence at display time from a fault model of the equipment. In the case of a decision tree, the complete tree is defined in the data. In the case of a dynamic fault isolation model, only the data needed to represent the fault model of the equipment is defined in the data.

Any of these diagnostic data structures can be described in terms of diagnostic tests ("test"), test outcomes ("outcome"), fault states ("fltstate"), repairable faults ("fault"), and fault rectification actions ("rect"). You begin a fault reporting or isolation process with a "test", which may be as simple as "what symptoms did you observe?", or as complex as a 50-step checkout procedure. Each "test" will have associated "outcomes" which associate possible test results with new fault states ("fltstate"). Test results are described as "precond" statements (e.g, voltage = 4.5v, light = dim, faultcode = A123, ...) which are asserted by the "test" procedure as the test is performed. The "outcome" elements relate those possible test results to fault states.

A "fltstate" state represents a node in a fault isolation decision tree or a set of plausible faults in a dynamic fault model. A "fltstate" state provides the information necessary to select the next diagnostic test. In a decision tree the test is explicitly identified by the "test" attribute of "fltstate". In a dynamic fault model, the "test" is not explicitly identified since the "test" attribute is empty, and the "fltstate" specifies a list of implicated faults ("impfault") and a list of exculpated faults ("expfault") for that state. Implicated faults are those states which are suspected to be bad in the fault state. Exculpated faults are those states known to be good in the fault state. These fault lists are then used by dynamic software to generate a list of appropriate "tests" which will further isolate the list of implicated faults. No matter how the test is selected, statically by the data or dynamically by the software, the selected "test" is performed and the process continues until a fault is isolated. In a decision tree, a fault is identified when you reach a final "fltstate" node which does not reference a "test", but lists the identified "fault". In a dynamic fault model, the final fault is identified by the software and is not explicitly represented by the "fltstate" element.

Once a "fault" is identified, the rectification or repair procedure ("rect") associated with the "fault" is performed. A rectification action also has an associated "test" which is generally a checkout task to verify that the rectification action successfully fixed the problem. The "rect" element also has a fault attribute which is a list of faults repaired by the rectification action.

Tests and rectifications can be performed by a human or machine agent. The elements "test" and "rect" have an "agent" attribute which states whether the action is performed by a human or a machine.

50.1.16 FAUL, FLTSTATE and RECT (See Diagram 50.2.16)

See above Section 50.1.15 above.

50.1.17 PARTINFO (See Diagram 50.2.17)

The elements "partinfo" and "partbase" define detailed parts information. "partinfo" describes an item by its reference designator ("refdes") which categorizes parts by their place in the system-subsystem hierarchy. "partinfo" describes the maintainer's view of the part information. Each "partinfo" element is related to a "partbase" which describes the item in terms of its part number ("partnum"). "partbase" describes the supply system's view of the part information. Several "partinfo" items could be related to the same "partbase".

50.1.18 PARTBASE (See Diagram 50.2.18)

See above Section 50.1.17 above.

50.1.19 CONTEXT (See Diagram 50.2.19)

Every CDM composite object also has context. Vehicle configuration, security level, and technician skill level are examples of context properties which determine the applicability of a particular data element to the situation at hand. "context" consists of a set of frequently used "effectively" attributes (security, ..., config, version), as well as a list of user defined "precond" (preconditions).

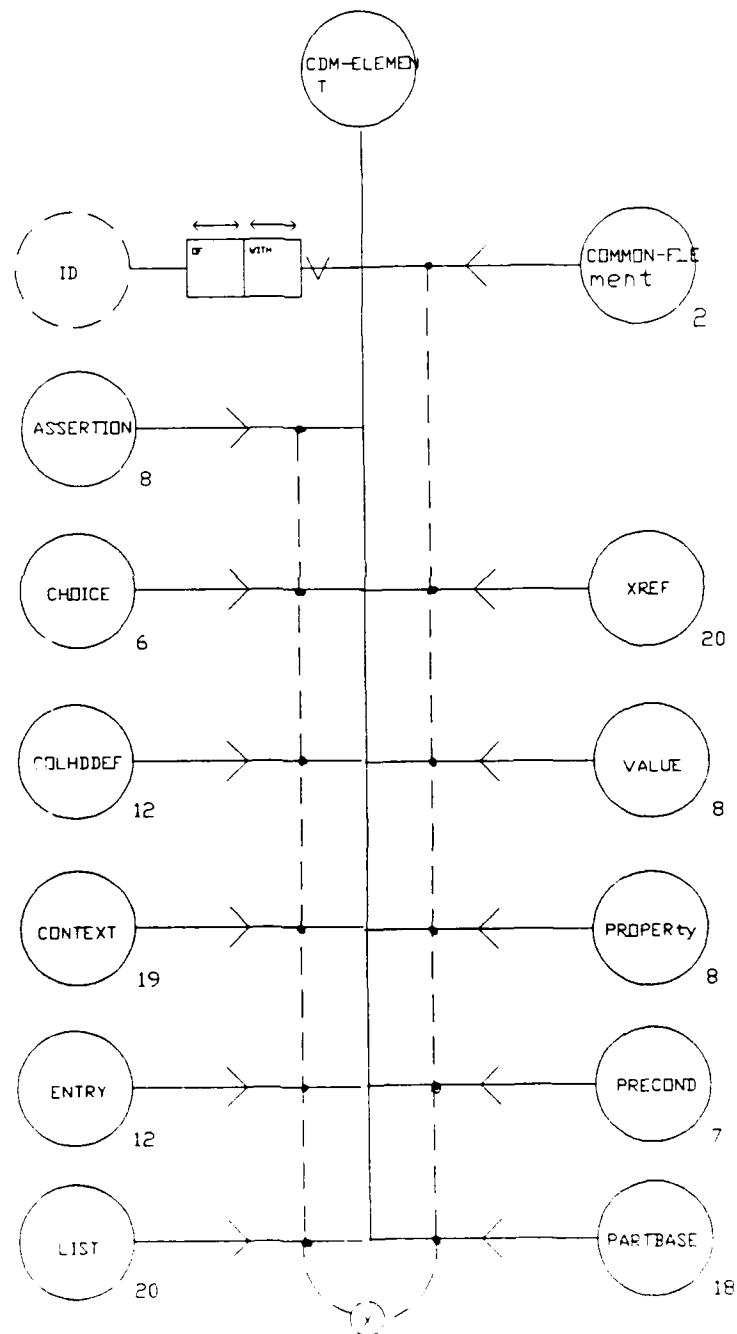
50.1.20 XREF and LIST (See Diagram 50.2.20)

The element "xref" defines a cross-reference or relational link. Each cross-reference has at least one internal reference or an external reference of the CDM. Internal references are represented by "elmtref" which is a reference id for any CDM element. External references are indicated by "exrefid" which is character data describing another file or database element. All cross references may have a type ("relation") which is a text string describing the nature of the reference (e.g., "theory", "IPB", "schematic", etc.). There is an optional attribute "attname" which may be used to narrow the "xref" to a particular attribute value of the cross-referenced element.

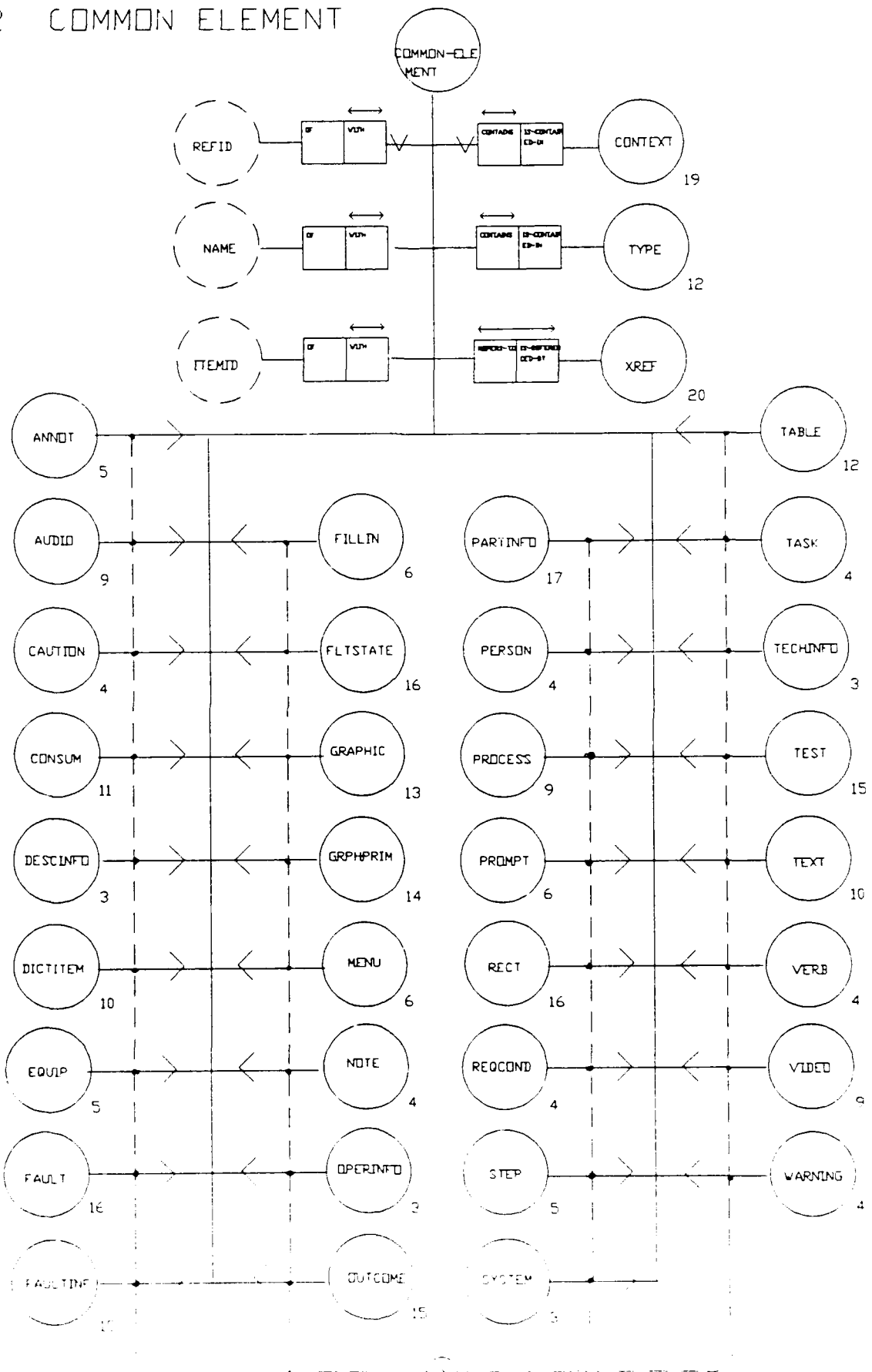
A "list" is a general purpose structure used to group individual elements into a list of elements which share a common context. For example, if you wanted to specify that a list of steps were all to be performed if a certain precondition were true, you could group those steps into a list with a single context which specified the desired precondition.

50.2 CDM/NIAM Conceptual Schema Diagrams.

50.2.1 CDM ELEMENT



50.2.2 COMMON ELEMENT



50.2.3 TECHINFO, SYSTEM, DESCINFO & OPERINFO

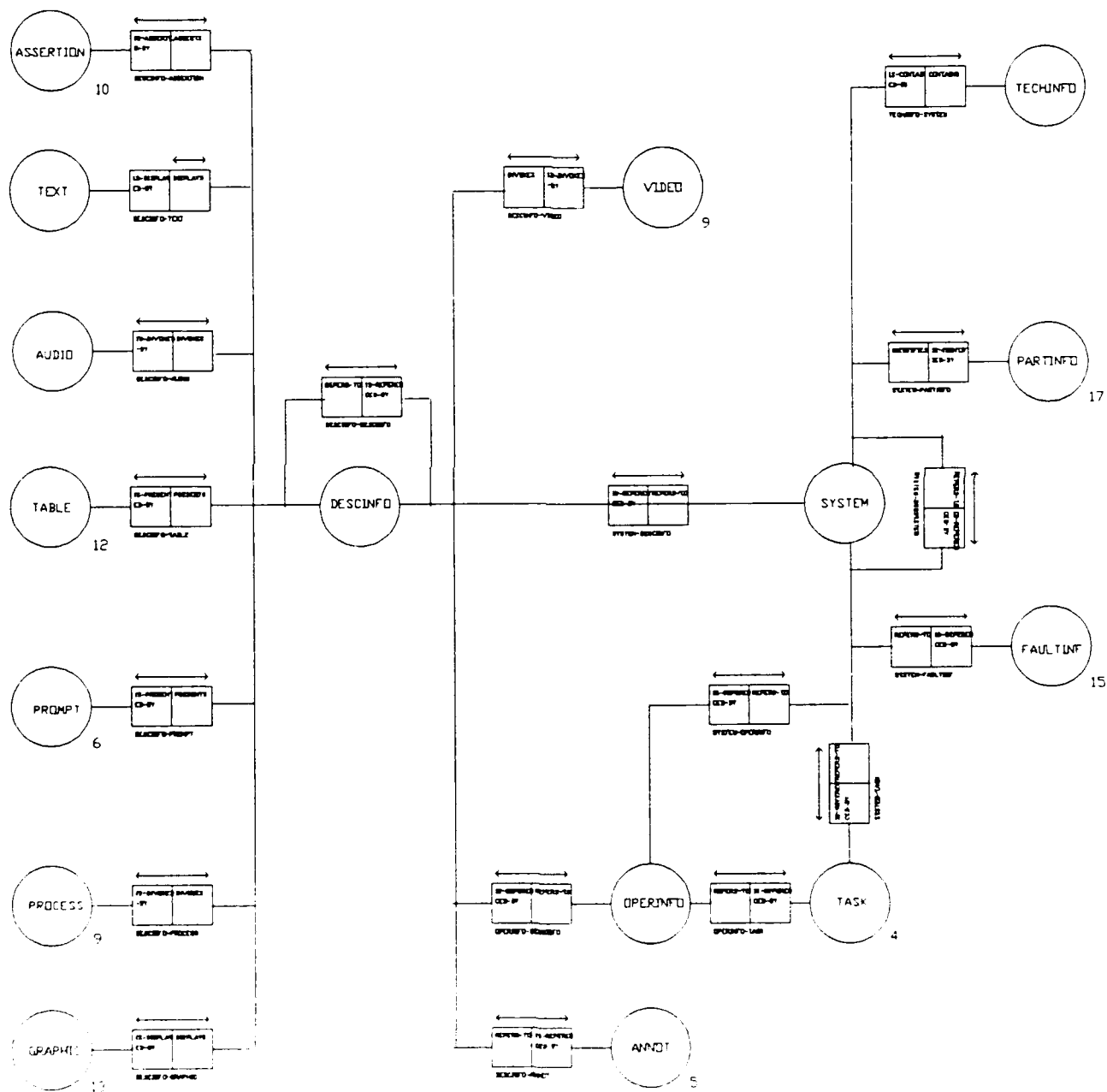
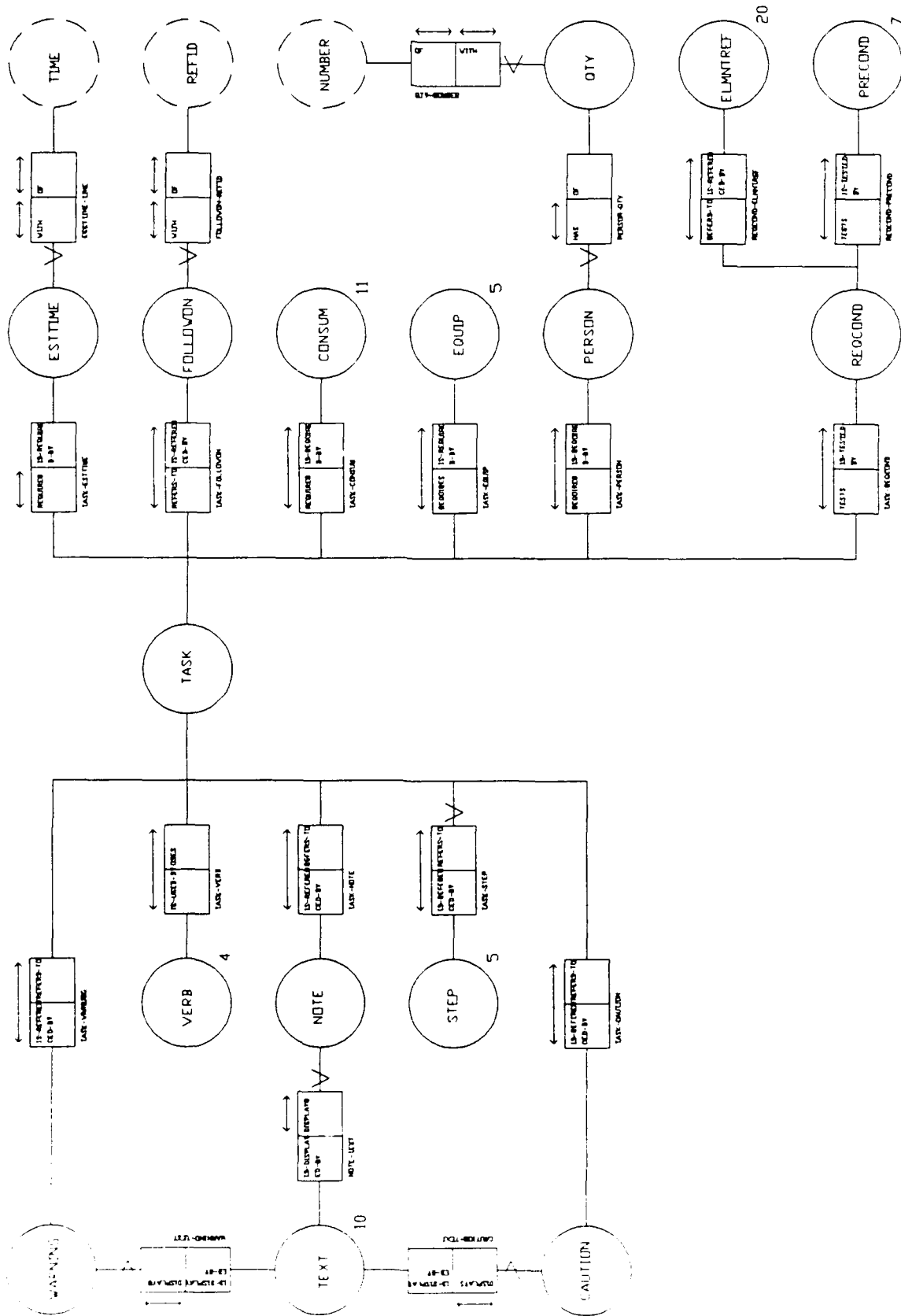
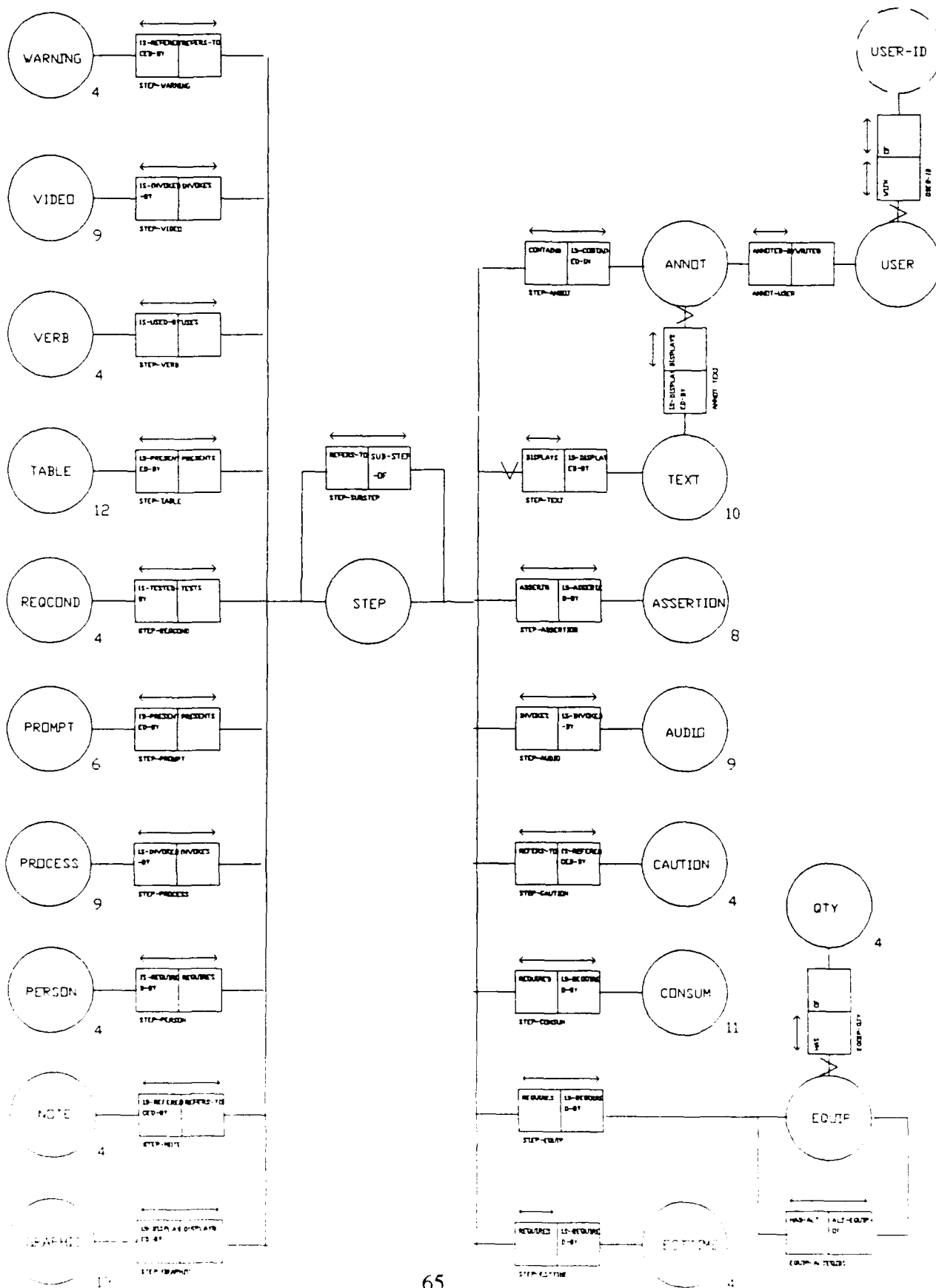


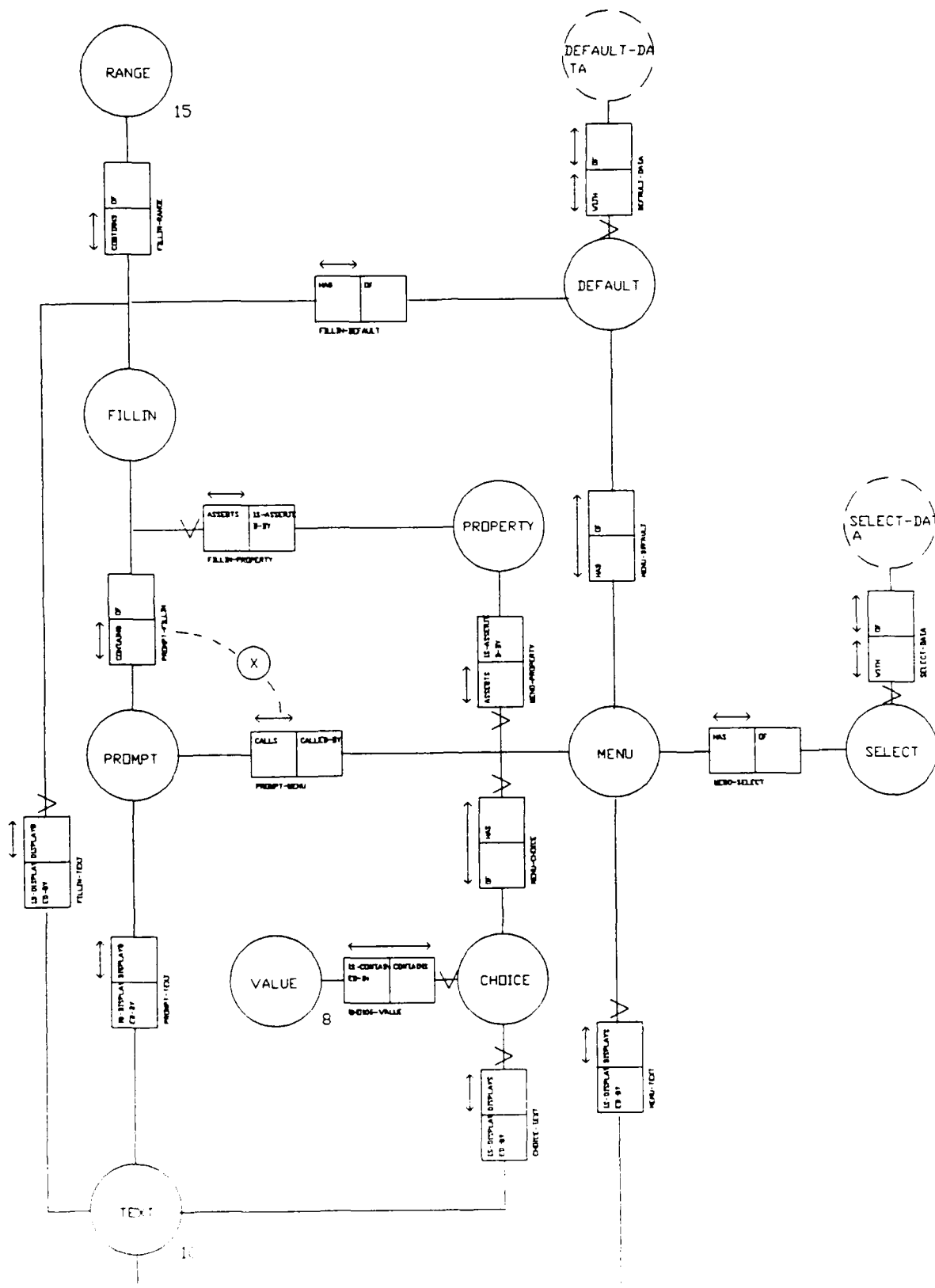
FIG 2-4 TASK, REQCOND, PERSON, CAUTION, NOTE & WARNING



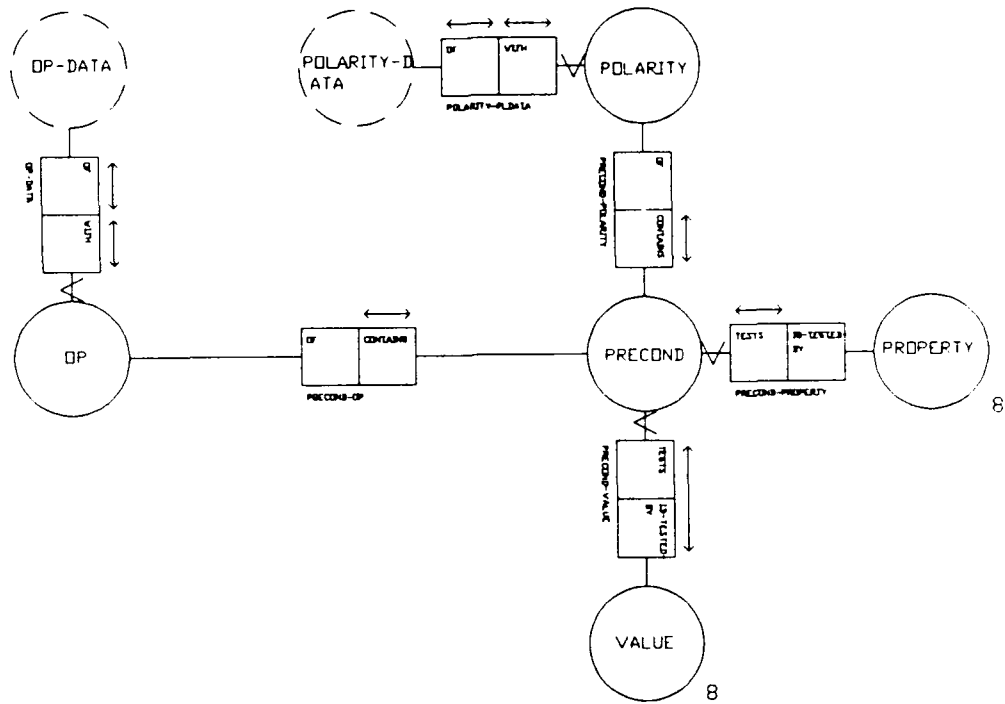
50.2.5 STEP, EQUIP, ANNOT & USER.



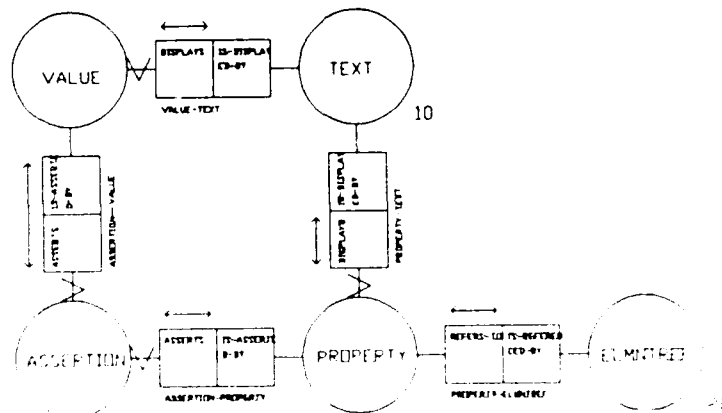
50.2.6 PROMPT, FILLIN, MENU & CHOICE.



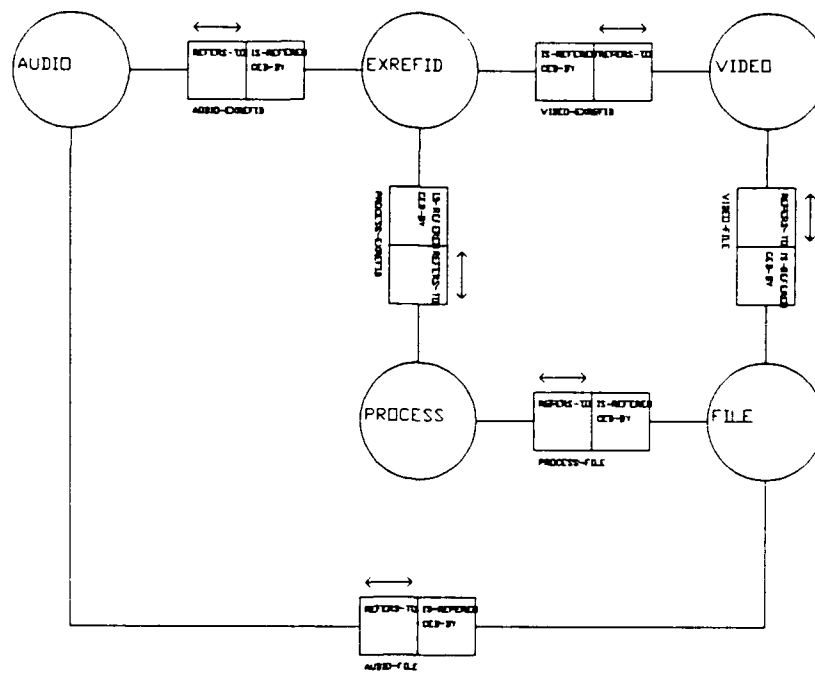
50.2.7 PRECOND



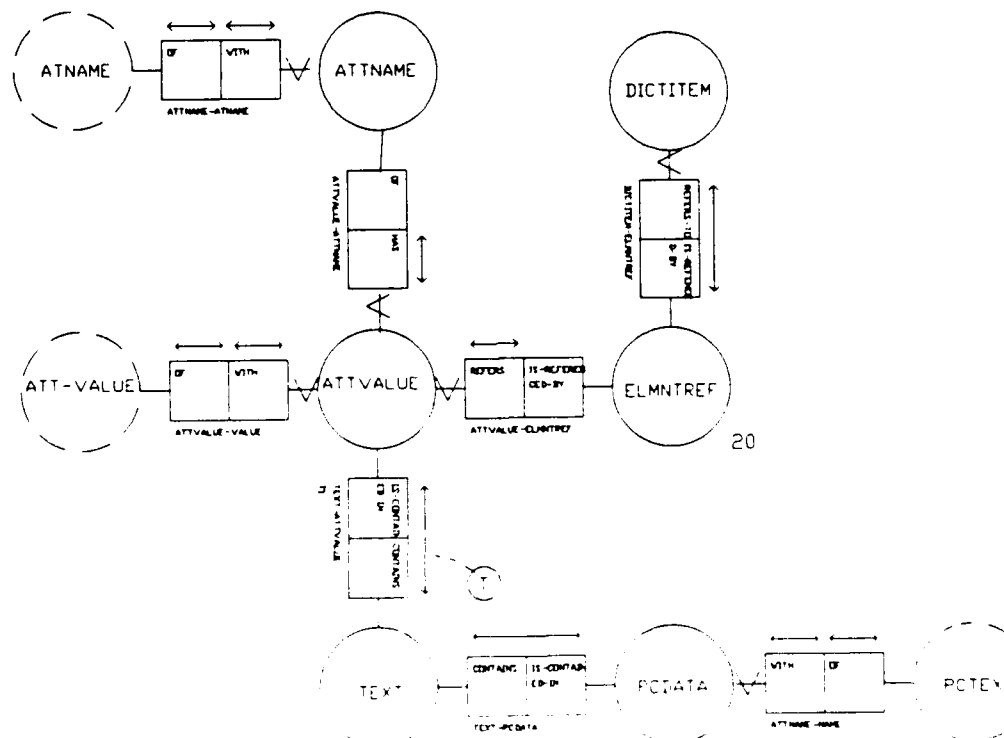
50.2.8 ASSERTION, PROPERTY & VALUE.



50.2.9 VIDEO, AUDIO & PROCESS.



50.2.10 TEXT & DICTITEM.



5.1.2.11 CONSUM.

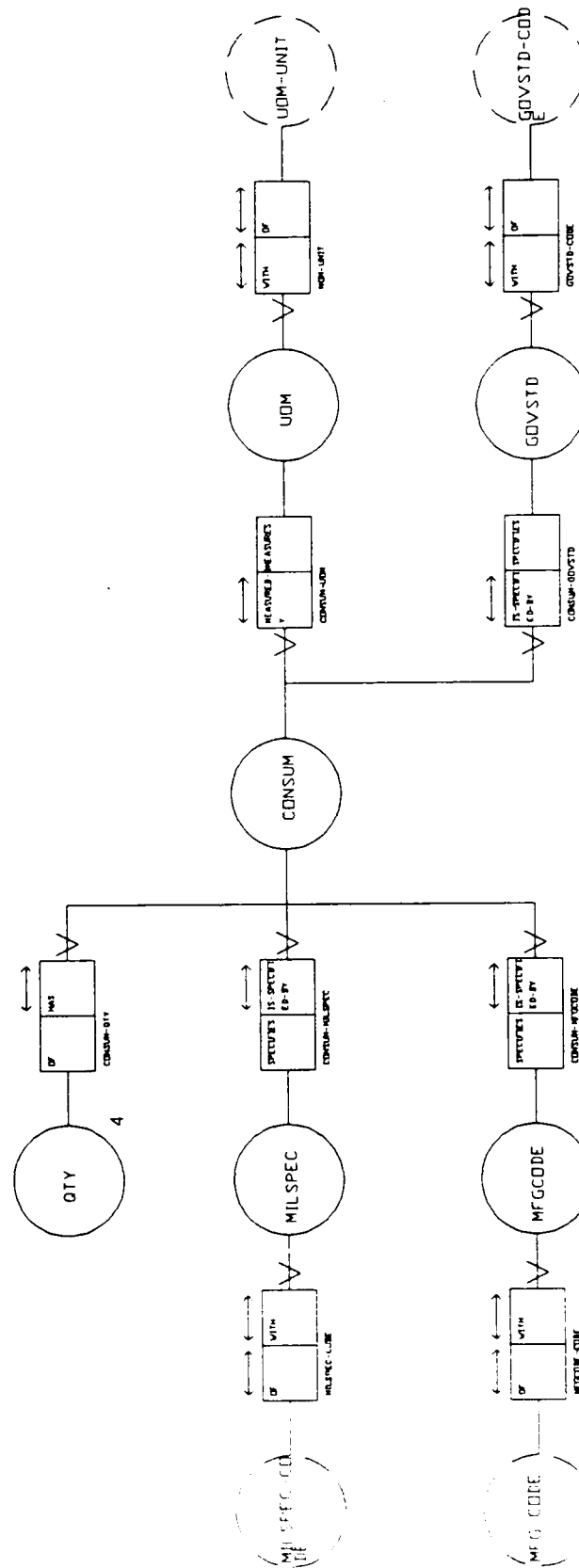
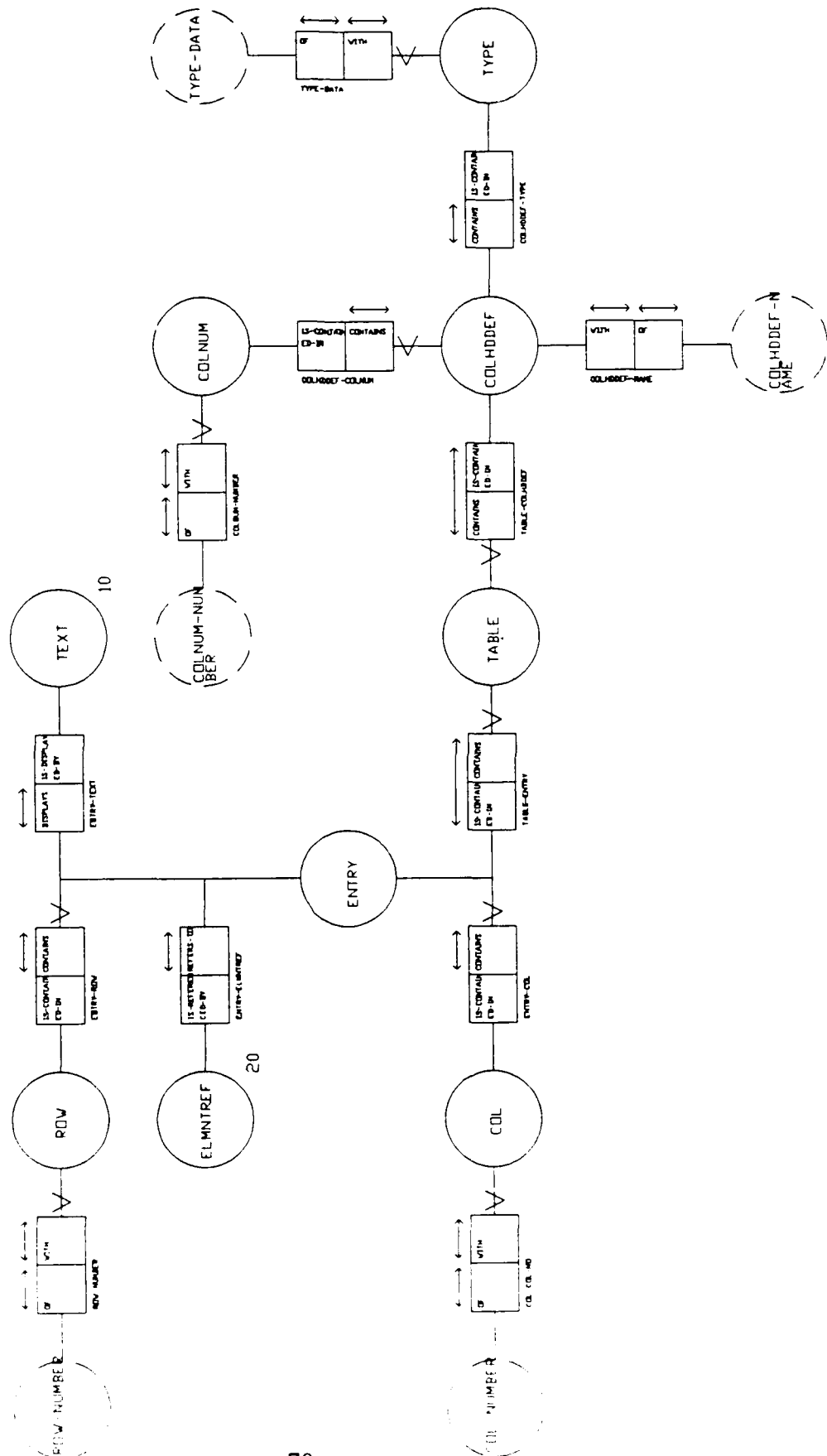
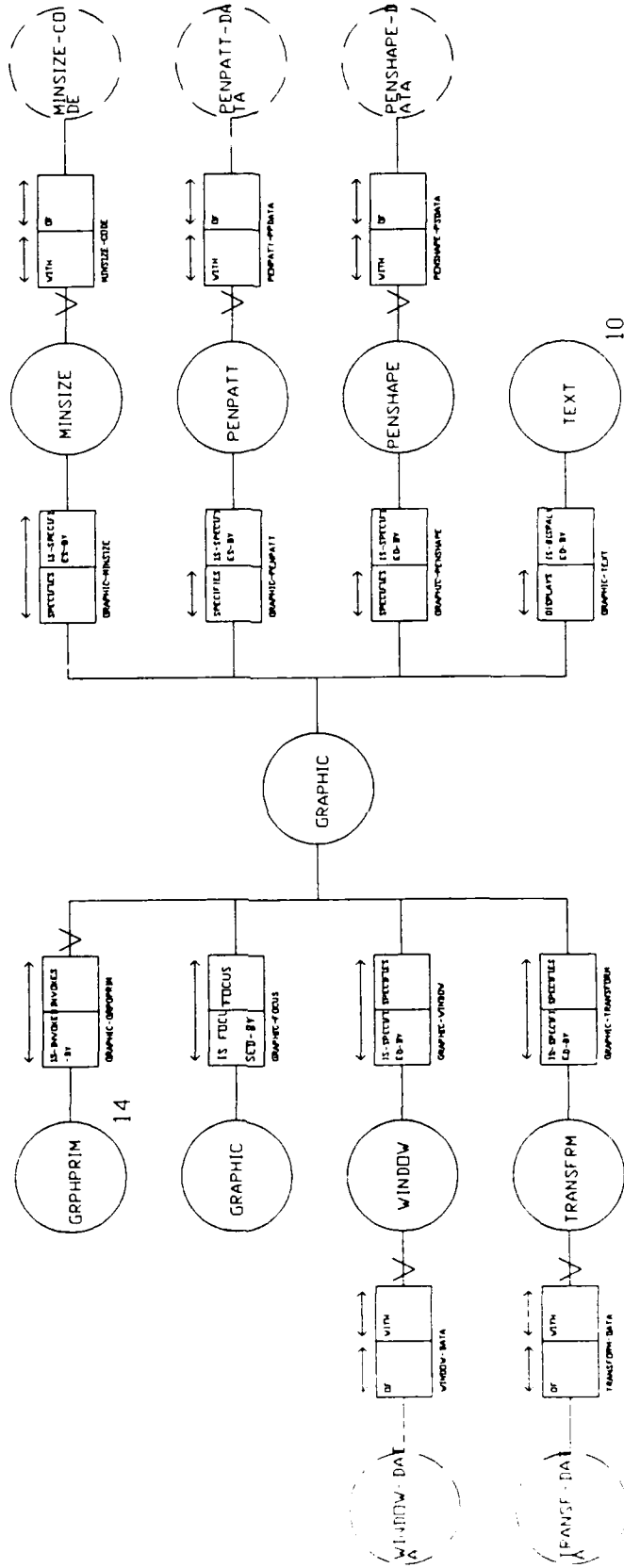


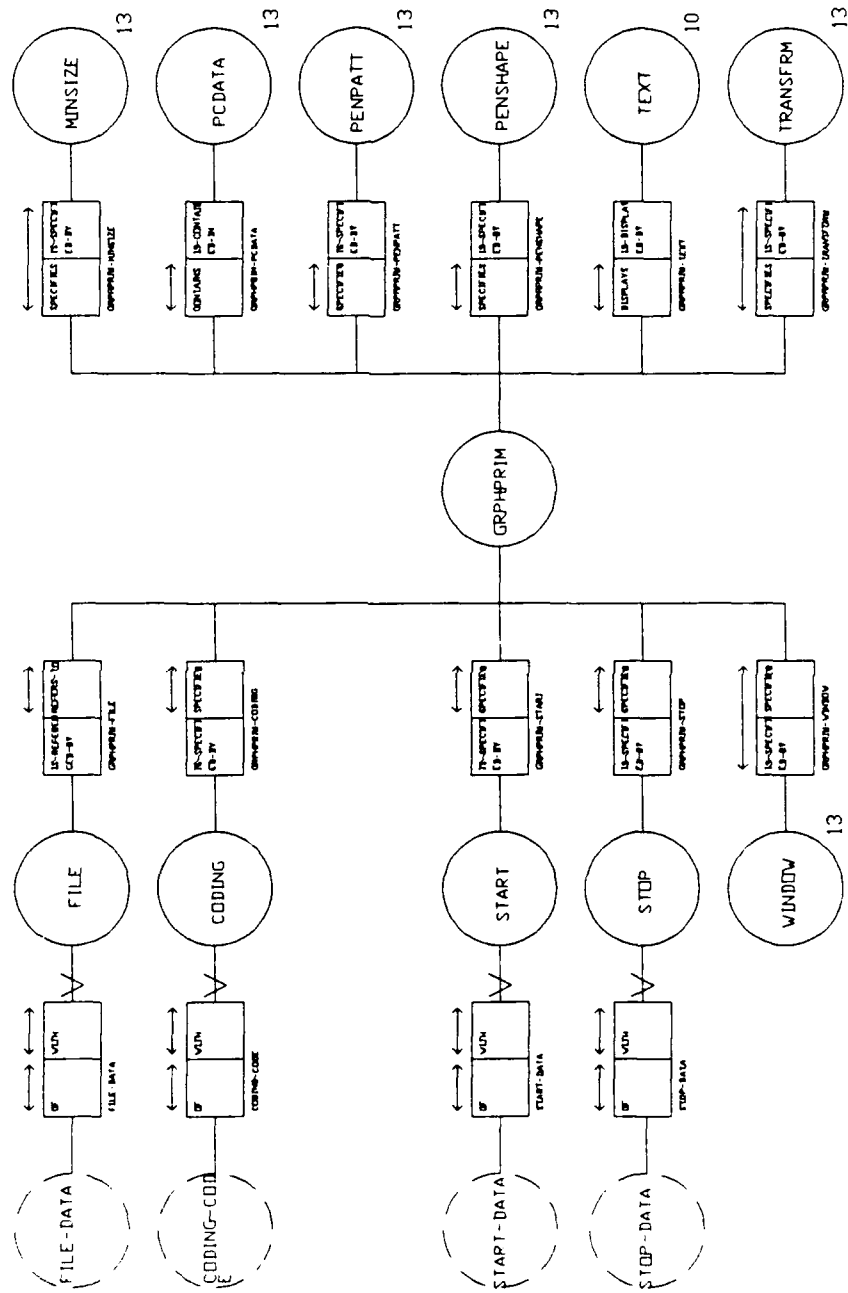
FIG. 12 TABLE, ENTRY & COLHDEF



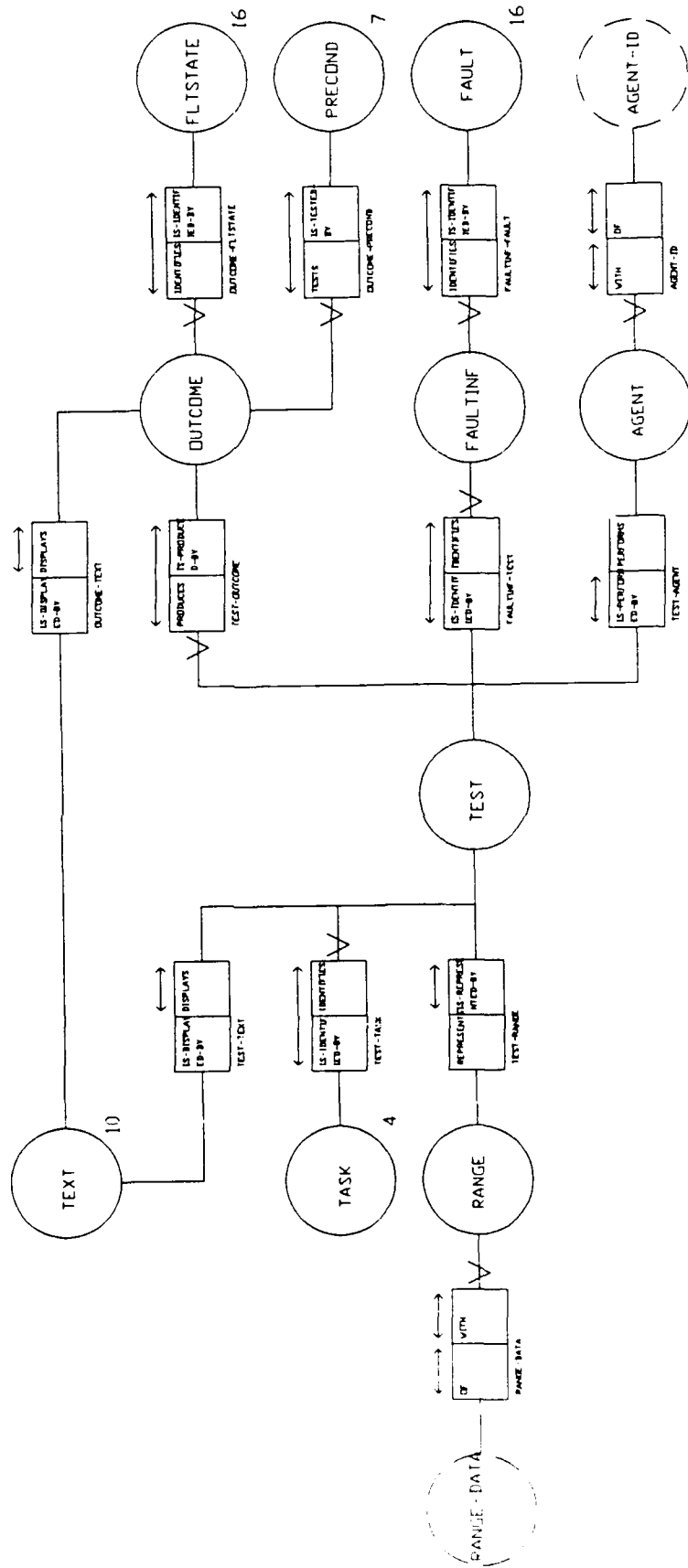
00.013 GRAPHIC



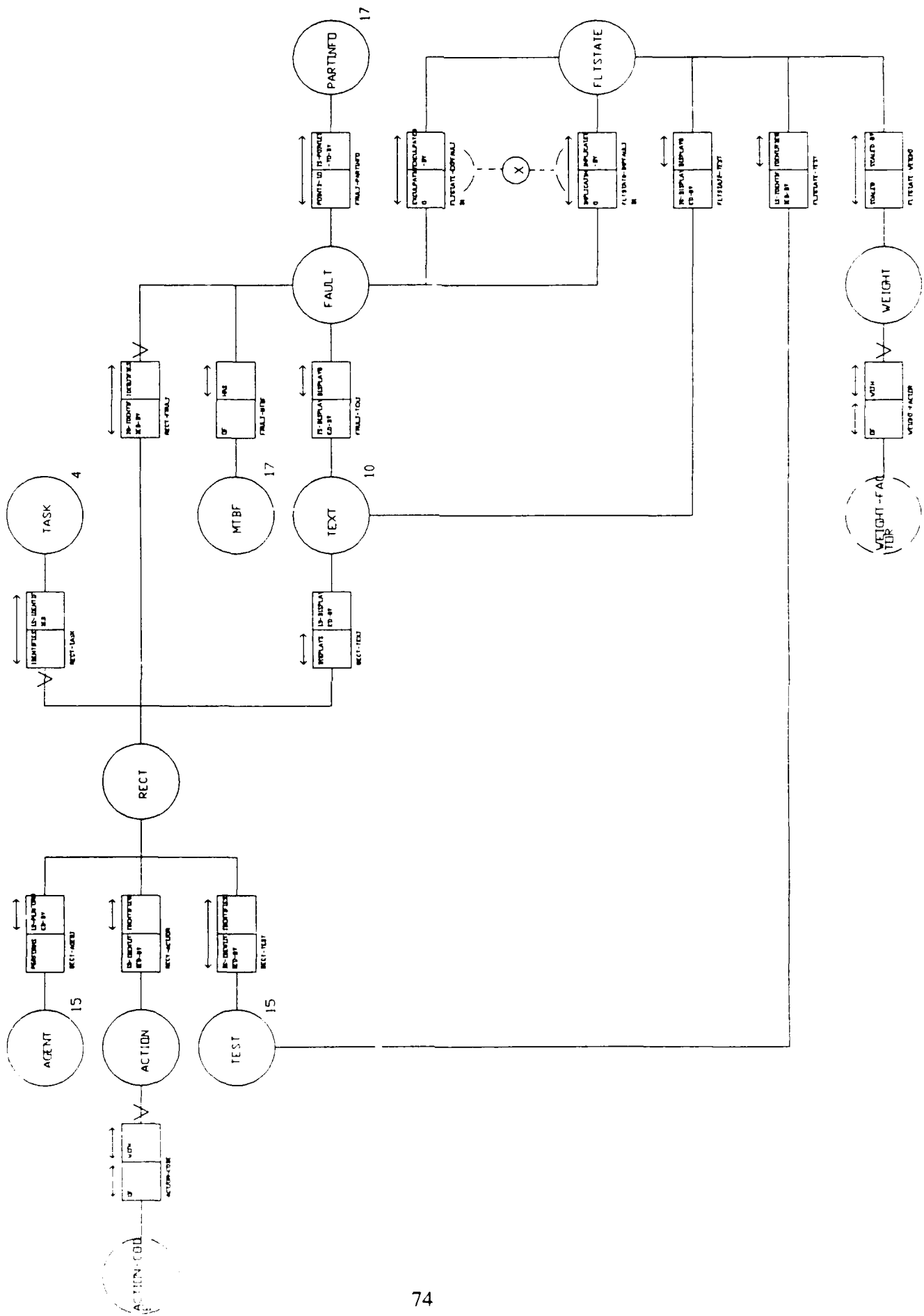
50.2.14 GRPHPRIM.



50.2.15 FAULTINFO, TEST & OUTCOME.



FAULT, FLTSTATE & RECT



SEP 17 PARTINFO

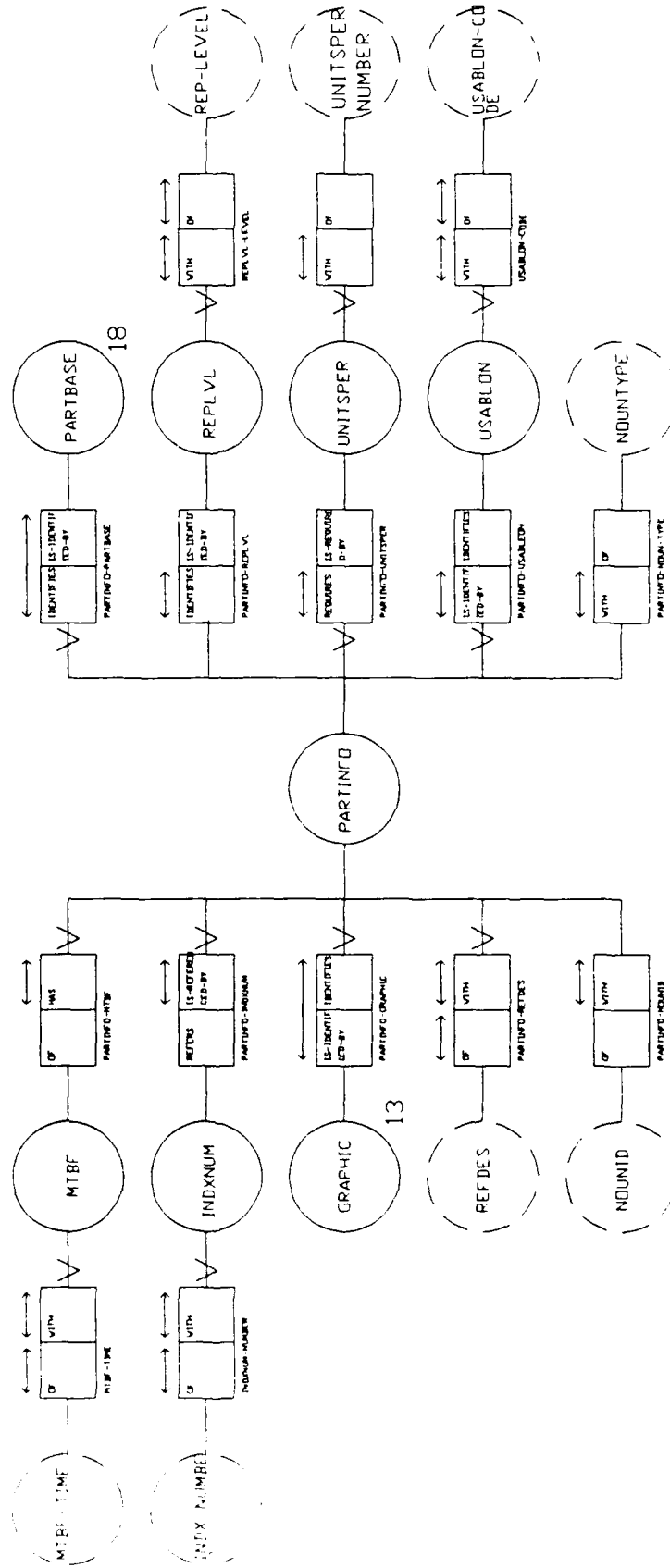
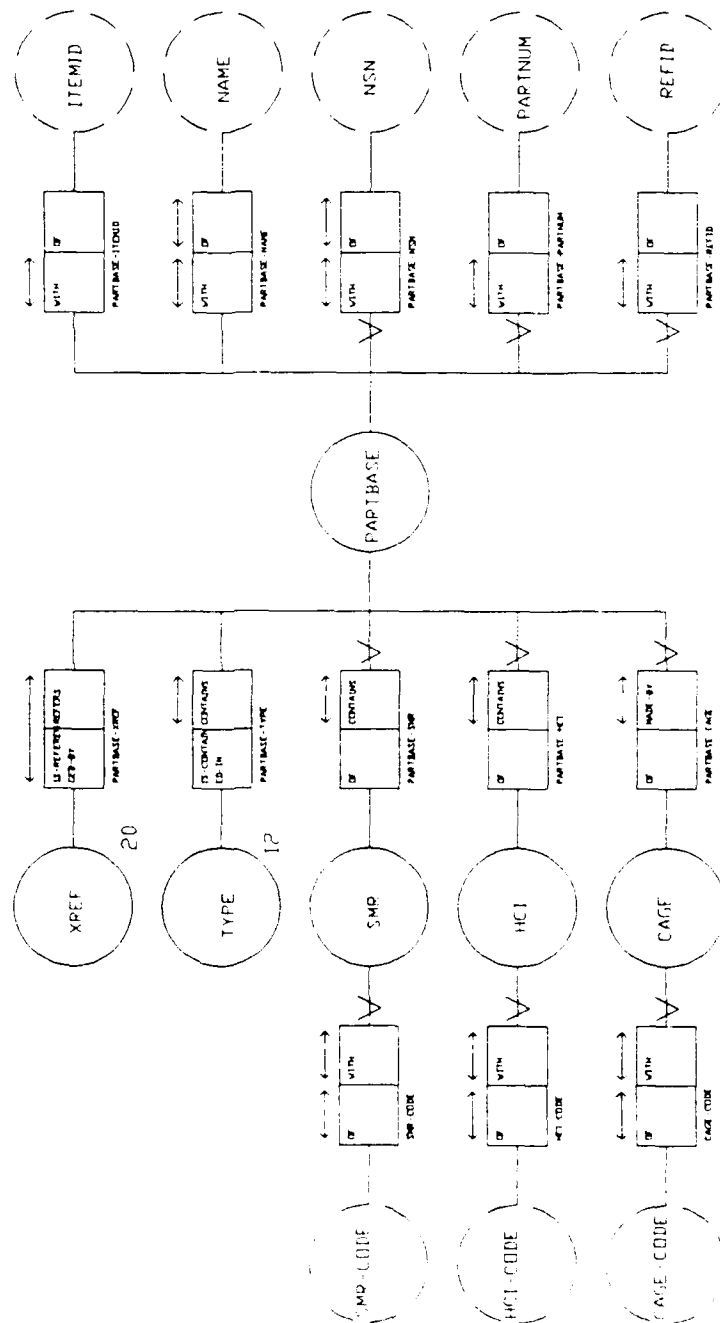
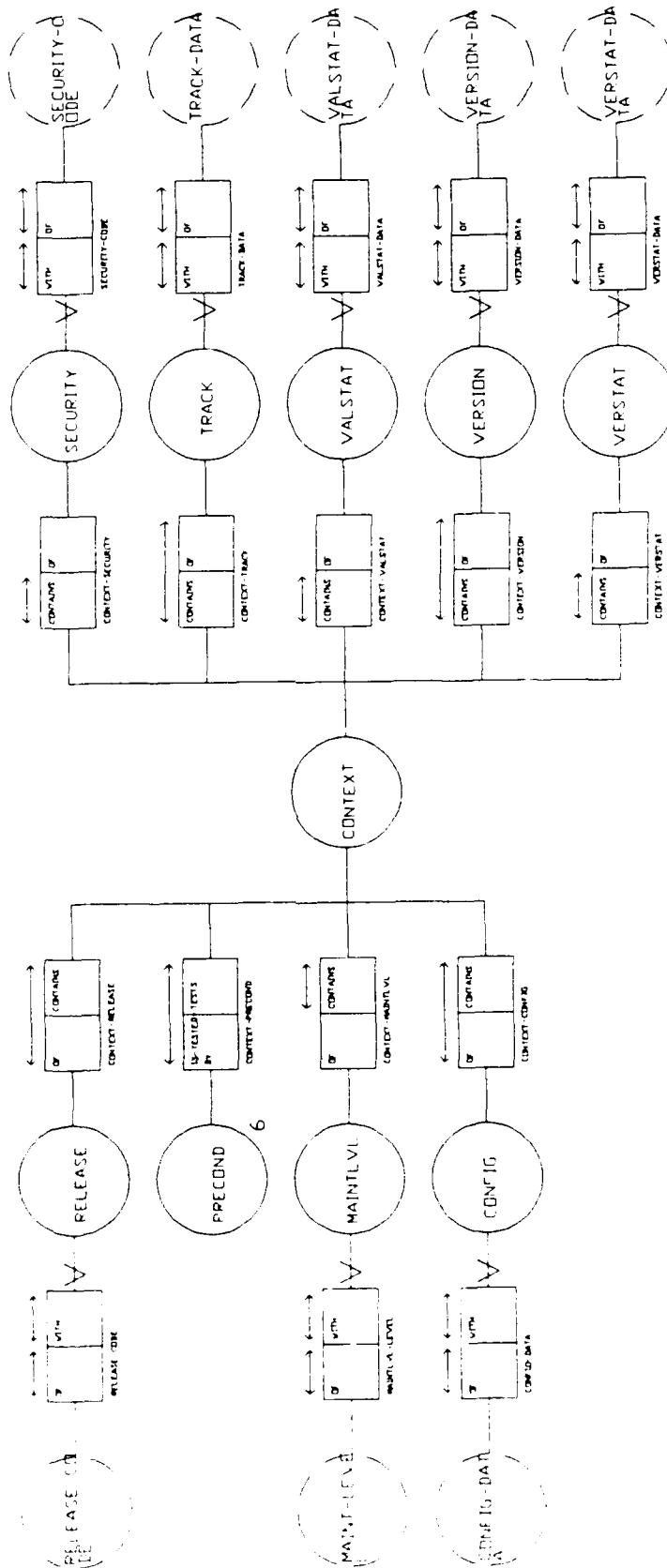


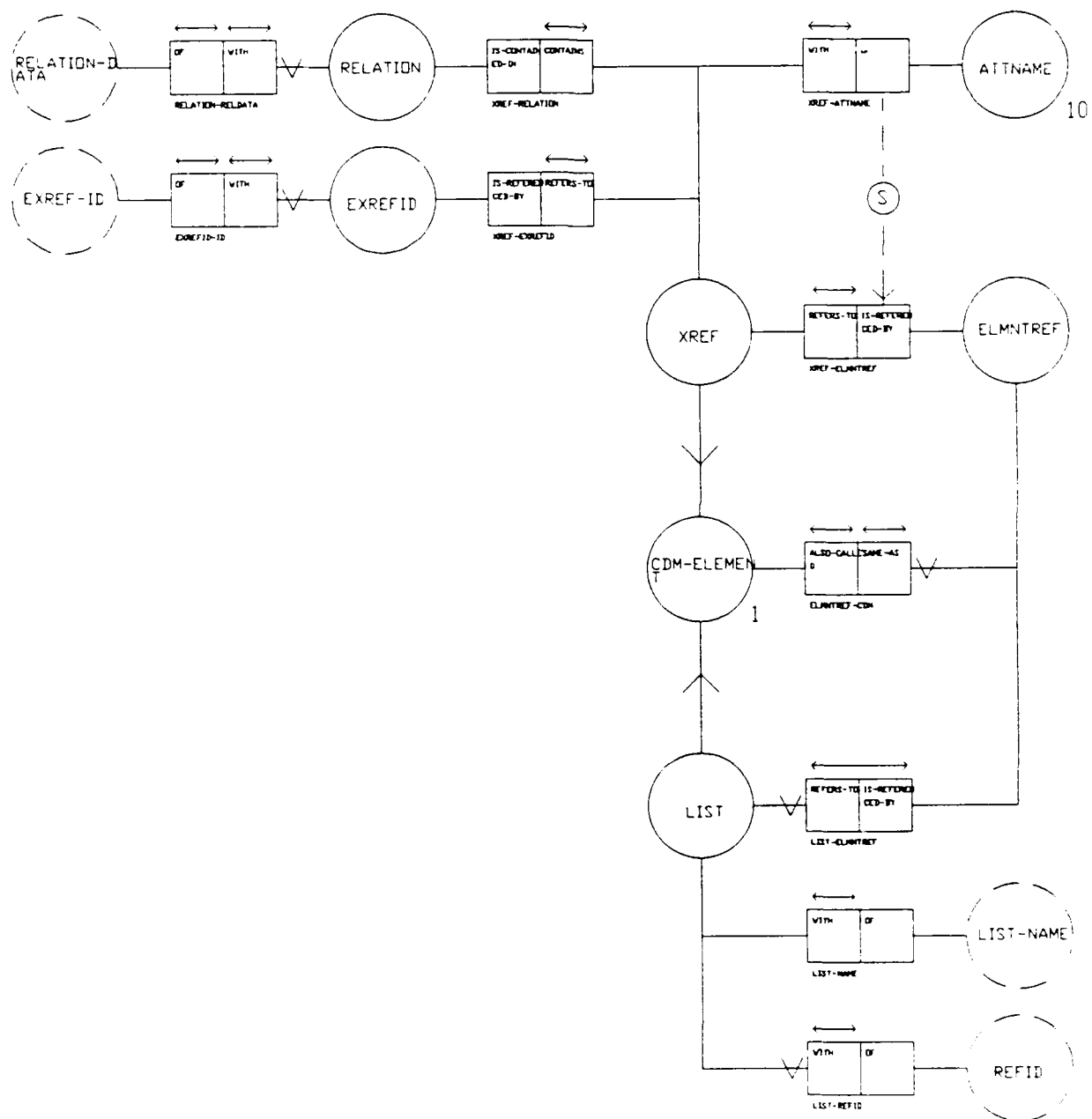
FIG. 18 PARTBASE.



CONTEXT - CONTEXT



50.2.20 XREF & LIST.



Appendix F - CDM/NIAM Data Dictionary

60.1 Scope: This Data Element Dictionary contains all the data elements (entities and attributes) of the CDM, and defines the relationships between them. The data elements in this Appendix are presented by the NIAM information analysis model in graphics form in Section 50.2 of Appendix E, interpreted in a neutral data model in a topographical form in Section 70.1 of Appendix G, presented in ORACLE-SQL database schema definition language in Section 80.1 of Appendix H, and finally expressed in Express information modeling language in Section 90 of Appendix I.

Each Data Element Definition (DED) entry contains some or all of the following items:

- a. Data Element Definition Number - DED Number.
- b. Data Element Definition Name - DED Name.
- c. Data Element Definition - DED.
- d. Data Element Field Format - Data format.
- e. Data Element Description - Description.
- f. Attribute of
- g. Entity Type Relationships/Constraints or
Label Type Descriptions/Constraints
- h. NIAM Diagram Number
- i. Example(s)

60.2 Data Element Dictionary Format. The general format for each entry of the Data Element Dictionary is as follows:

DED Number DED Name - DED

 Data Format:

 Description:

 Attribute of:

 Entity Type Relationships/Constraints: NIAM Diagram Number
 or Label Type Descriptions/Constraints:

 Example(s):

The terms used in the general format description above are defined below:

60.2.1 DED Number: A sequentially assigned number used to locate and reference each data element in the dictionary.

60.2.2 DED Name: The term used to uniquely identify the data element.

60.2.3 DED: A short phrase used to define and describe the data element.

60.2.4 Data Format (for label type only): Specifies the length type.

positional justification, and decimal placement for a data element field or subfield thereof as described below:

a. Length: The number of maximum characters is specified.

b. Type: Specifies the character type, wherein:

A - all characters are alphabetical.

N - all characters are numerical.

X - characters are alphanumeric, special or any combination.

D - characters are real numbers in floating decimal format.

c. Justification: Specifies from which side of the field the data element characters are entered. For example, those starting at the left are left justified (L).

L - left justified.

R - right justified.

F - fixed, occupying the entire field.

d. Decimal Placement: Specifies the number of character positions to the right of the assumed decimal point when the data element characters are all numerical. A dash (-) is used if this column is not applicable.

e. NARRATIVE Field format: An extended narrative data field holding a maximum of 65,536 text characters.

60.2.5 Data Element Description: A narrative definition of the data element in sufficient detail to present a clear and complete understanding of the precise data or element of information that the data element represents.

60.2.6 Attribute of: Identifies those entities with which this data element is an attribute of. The attribute represents a property, or characteristic of the entity.

60.2.7 Entity Type Relationships/Constraints or Label Type Descriptions/Constraints - Specifies whether the data element is of entity type or label type. For detailed definitions of entity type, label type, as well as relationships and constraints, see Appendix C.

The "sentences" which follow the "Entity Type Relationships/Constraints" or "Label Type Descriptions/Constraints" headings in Section 3.2 are generated by the IAST software tool.

60.2.8 NIAM Diagram Number: Specifies NIAM diagrams in Appendix C, Section 30.2 in which the data element appears.

60.2.9 Example(s): Provides relevant examples.

60.3 Data Element Definition:

001 ACTION - Maintenance Fault Rectification Action

Description: Describes the type of maintenance action required to rectify or fix a fault. The action can be a "SWAP" which means it is a removal/replacement action, or it can be a "MAINT" action which means it is an adjustment, alignment, etc. action.

Attribute of: RECT

Entity Type Relationships/Constraints: See Section 50.2.16

An ACTION is an entity type.

An ACTION may be IDENTIFIED-BY any number of RECT's.

An ACTION must be WITH exactly one ACTION-CODE.

002 ACTION-CODE - A Label Type of ACTION

Data Format: 5 A L

Label Type Descriptions/Constraints: See Section 50.2.16

An ACTION-CODE is a label type with representation Character 5.

An ACTION-CODE may be OF at most one ACTION.

Example(s): "swap" or "maint"

003 AGENT - Performer Of Fault Isolation

Description: Designates whether a maintenance action (e.g. a test or a rectification action) is performed by a human or machine.

Attribute of: RECT, TEST

Entity Type Relationships/Constraints: See Section 50.2.15

An AGENT is an entity type.

An AGENT may PERFORM any number of RECT's.

An AGENT may PERFORM any number of TEST's.

An AGENT must be WITH exactly one AGENT-ID.

004 AGENT-ID - A Label Type of AGENT

Data Format: 7 A L

Label Type Descriptions/Constraints: See Section 50.2.15

An AGENT-ID is a label type with representation Character 7.

An AGENT-ID may be OF at most one AGENT.

Example(s): "human" or "machine"

005 ANNOT - User Annotation

Description: References a comment or note from the technician.

Annotations can refer to descriptive information or a specific step within a procedural task.

Attribute of: DESCINFO, STEP

Entity Type Relationships/Constraints: See Section 50.2.5

An ANNOT is an entity type.

An ANNOT is a subtype of COMMON-ELEMENT.

An ANNOT may be ANNOTED-BY at most one USER.

An ANNOT must DISPLAY exactly one TEXT.

An ANNOT may be CONTAINED-IN any number of STEP's.

An ANNOT may be REFERENCED-BY any number of DESCINFO's.

006 ASSERTION - Assertion

Description: An assertion is a fact defined in terms of a PROPERTY-VALUE pair which expresses an author defined property to be "asserted" as the system processes the STEP or DESCINFO in question.

Attribute of: DESCINFO, STEP

Entity Type Relationships/Constraints: See Section 50.2.8

An ASSERTION is an entity type.

An ASSERTION is a subtype of CDM-ELEMENT.

An ASSERTION must ASSERT exactly one PROPERTY.

An ASSERTION must ASSERT one or more VALUE's.

An ASSERTION may be ASSERTED-BY any number of DESCINFO's.

An ASSERTION may be ASSERTED-BY any number of STEP's.

007 ATNAME - A Label Type of ATNAME

Data Format: 80 A L

Label Type Descriptions/Constraints: See Section 50.2.10

An ATNAME is a label type with representation Character 80.

An ATNAME may be OF at most one ATNAME.

008 ATT-VALUE - A Label Type of ATTVALUE

Data Format: 80 A L

Label Type Descriptions/Constraints: See Section 50.2.10

An ATT-VALUE is a label type with representation Character 80.

An ATT-VALUE may be OF at most one ATTVALUE.

009 ATTNAME - Attribute Name

Description: Represents an attribute value of a cross-referenced element so as to "focus" the scope of the reference.

Attribute of: ATTVALUE, XREF

Entity Type Relationships/Constraints: See Section 50.2.10

An ATTNNAME is an entity type.

An ATTNNAME may be OF any number of ATTVALUE's.

An ATTNNAME may be OF any number of XREF's.

An ATTNNAME must be WITH exactly one ATNAME.

010 ATTVALUE - Attribute Value

Description: Identifies the values of other CDM element attributes. This will allow a standard terminology to be referenced consistently throughout the database, as a result changes to standard terms can be made in one location and automatically be updated throughout the database.

Attribute of: TEXT

Entity Type Relationships/Constraints: See Section 50.2.10

An ATTVALUE is an entity type.

An ATTVALUE must HAVE exactly one ATTNNAME.

An ATTVALUE may be CONTAINED-IN any number of TEXT's.

An ATTVALUE must be REFERRED to exactly one ELMNTREF.

An ATTVALUE must be WITH exactly one ATT-VALUE.

011 AUDIO - Audio Data

Description: References an external audio sequence which may be represented as a file or some other external reference.

Attribute of: DESCINFO, STEP

Entity Type Relationships/Constraints: See Section 50.2.9

An AUDIO is an entity type.

An AUDIO is a subtype of COMMON-ELEMENT.

An AUDIO may be INVOKED-BY any number of DESCINFO's.

An AUDIO may be INVOKED-BY any number of STEP's.

An AUDIO may REFER-TO at most one EXREFID.

An AUDIO may REFER-TO at most one FILE.

012 CAGE - Commercial and Government Entity

Description: Represents a commercial and government entity number. A five digit number assigned by the Defense Logistic Services Center (DLSC) to the design control activity or actual manufacturer of an item contained in the Cataloguing Handbook H4/H8 series.

Attribute of: PARTBASE

Entity Type Relationships/Constraints: See Section 50.2.18

A CAGE is an entity type.

A CAGE may be OF any number of PARTBASE's.

A CAGE must be WITH exactly one CAGE-CODE.

013 CAGE-CODE - A Label Type of CAGE

Data Format: 5 A F

Label Type Descriptions/Constraints: See Section 50.2.18

A CAGE-CODE is a label type with representation Character 5.

A CAGE-CODE may be OF at most one CAGE.

014 CAUTION - Caution

Description: References a caution that is associated with a particular task or step. A caution notifies the technician that the task or step may harm the equipment if not properly performed.

Attribute of: STEP, TASK

Entity Type Relationships/Constraints: See Section 50.2.4

A CAUTION is an entity type.

A CAUTION is a subtype of COMMON-ELEMENT.

A CAUTION must DISPLAY exactly one TEXT.

A CAUTION may be REFERENCED-BY any number of STEP's.

A CAUTION may be REFERENCED-BY any number of TASK's.

015 CDM-ELEMENT - CDM Element

Description: Represents the root of the CDM element hierarchical structure.

Attribute of: None

Entity Type Relationships/Constraints: See Section 50.2.1

A CDM-ELEMENT is an entity type.

A CDM-ELEMENT may be ALSO-CALLED at most one ELMNTREF.

A CDM-ELEMENT must be WITH exactly one ID.

The following subtypes of the CDM-ELEMENT are mutually exclusive: ASSERTION, CHOICE, COLHDDEF, COMMON-ELEMENT, CONTEXT, ENTRY, LIST, PARTBASE, PRECOND, PROPERTY, VALUE, and XREF.

016 CHOICE - Choice Of A Menu

Description: References the text to be displayed to the user in the list of possible selections within a menu.

Attribute of: MENU

Entity Type Relationships/Constraints: See Section 50.2.6

A CHOICE is an entity type.

A CHOICE is a subtype of CDM-ELEMENT.

A CHOICE must CONTAIN one or more VALUE's.

A CHOICE must DISPLAY exactly one TEXT.

A CHOICE may be OF any number of MENU's.

017 CODING - Graphics Codes

Description: References the particular type of graphic file (e.g. IGES, CGM, etc.) that the current graphic is stored.

Attribute of: GRPHPRIM

Entity Type Relationships/Constraints: See Section 50.2.14

A CODING is an entity type.

A CODING may be SPECIFIED-BY any number of GRPHPRIM's.

A CODING must be WITH exactly one CODING-CODE.

018 CODING-CODE - A Label Type of CODING

Data Format: 80 X L

Label Type Descriptions/Constraints: See Section 50.2.14

A CODING-CODE is a label type with representation Alphanumeric 80.

A CODING-CODE may be OF at most one CODING.

Example(s): "cgmchar", "cgmbin", "cgmclean", "fax", "iges", "dxf" or "gks".

019 COL - Table Entry Column

Description: Represents the specific column of a table where the current entry belongs.

Attribute of: ENTRY

Entity Type Relationships/Constraints: See Section 50.2.12

A COL is an entity type.

A COL may be CONTAINED-IN any number of ENTRY's.

A COL must be WITH exactly one COL-NUMBER.

020 COL-NUMBER - A Label Type of COL

Data Format: 4 N R

Label Type Descriptions/Constraints: See Section 50.2.12

A COL-NUMBER is a label type with representation Numeric 4.

A COL-NUMBER may be OF at most one COL.

021 COLHDDEF - Column Heading Definition.

Description: Identifies the column heading definitions portion of a table.

Attribute of: TABLE

Entity Type Relationships/Constraints: See Section 50.2.12

A COLHDDEF is an entity type.

A COLHDDEF is a subtype of CDM-ELEMENT.

A COLHDDEF must CONTAIN exactly one COLNUM.

A COLHDDEF may CONTAIN at most one TYPE.

A COLHDDEF may be CONTAINED-IN any number of TABLE's.

A COLHDDEF must have exactly one COLHDDEF-NAME.

022 COLHDDEF-NAME - A Label Type of COLHDDEF

A COLHDDEF-NAME is a label type with representation Alphanumeric 20.

Label Type Descriptions/Constraints: See Section 50.2.12

A COLHDDEF-NAME is a label type with representation Alphanumeric 20.

A COLHDDEF-NAME may be OF at most one COLHDDEF.

023 COLNUM - Column of a Table

Description: Represents the column number(s) within the table to which the current element refers.

Attribute of: COLHDDEF

Entity Type Relationships/Constraints: See Section 50.2.12

A COLNUM is an entity type.

A COLNUM may be CONTAINED-IN any number of COLHDDEF's.

A COLNUM must be WITH exactly one COLNUM-NUMBER.

024 COLUMN-NUMBER - A Label Type of COLUMN

Data Format: 2 N R

Label Type Descriptions/Constraints: See Section 50.2.12

A COLNUM-NUMBER is a label type with representation Numeric 2.

A COLNUM-NUMBER may be OF at most one COLNUM.

025 COMMON-ELEMENT - Common Element

Description: Specifies those elements (entity types) which share the following entity/label types: REFID, NAME, ITEMID, TYPE, CONTEXT, and XREF.

Attribute of: None

Entity Type Relationships/Constraints: See Section 50.2.2

A COMMON-ELEMENT is an entity type.

A COMMON-ELEMENT is a subtype of CDM-ELEMENT.

A COMMON-ELEMENT must CONTAIN exactly one CONTEXT.

A COMMON-ELEMENT may CONTAIN at most one TYPE.

A COMMON-ELEMENT may REFER-TO any number of XREF's.

A COMMON-ELEMENT may be WITH at most one ITEMID.

A COMMON-ELEMENT may be WITH at most one NAME.

A COMMON-ELEMENT must be WITH exactly one REFID.

The following subtypes of the COMMON-ELEMENT are mutually exclusive: ANNOT, AUDIO, CAUTION, CONSUM, DESCINFO, DICTITEM, EQUIP, FAULT, FAULTINF, FILLIN, FLTSTATE, GRAPHIC, GRPHPRIM, MENU, NOTE, OPERINFO, OUTCOME, PARTINFO, PERSON, PROCESS, PROMPT, RECT, REQCOND, STEP, SYSTEM, TABLE, TASK, TECHINFO, TEST, TEXT, VERB VIDEO, and WARNING.

026 CONFIG - Configuration.

Description: Represents the configuration of the current system being maintained. It allows the system to automatically choose the correct path with regard to the configuration which has been entered into the system. It based on effectivity codes in current technical orders.

Attribute of: CONTEXT

Entity Type Relationships/Constraints: See Section 50.2.19

A CONFIG is an entity type.

A CONFIG may be OF any number of CONTEXT's.

A CONFIG must be WITH exactly one CONFIG-DATA.

027 CONFIG-DATA

Data Format: 20 X L

Label Type Descriptions/Constraints: See Section 50.2.19

A CONFIG-DATA is a label type with representation Alphanumeric 20.

A CONFIG-DATA may be OF at most one CONFIG.

028 CONSUM - Consumable Supply

Description: Identifies information about a consumable required for a particular STEP.

Attribute of: STEP, TASK

Entity Type Relationships/Constraints: See Section 50.2.11

A CONSUM is an entity type.

A CONSUM is a subtype of COMMON-ELEMENT.

A CONSUM must HAVE exactly one QTY.

A CONSUM may be REQUIRED-BY any number of STEP's.

A CONSUM may be REQUIRED-BY any number of TASK's.

A CONSUM must BE-SPECIFIED-BY exactly one GOVSTD.

A CONSUM must BE-SPECIFIED-BY exactly one MFGCODE.

A CONSUM must BE-SPECIFIED-BY exactly one MILSPEC.

A CONSUM must BE-MEASURED-BY exactly one UOM.

029 CONTEXT - Effectivity Analysis

Description: References both a set of frequently used effectivity

attributes and a list of user defined preconditions to ensure a proper transversal of database information to provide the correct information to the technician for the current system being maintained/repaired.

Attribute of: COMMON-ELEMENT

Entity Type Relationships/Constraints: See Section 50.2.19

- A CONTEXT is an entity type.
- A CONTEXT is a subtype of CDM-ELEMENT.
- A CONTEXT may CONTAIN any number of CONFIG's.
- A CONTEXT may CONTAIN at most one MAINTLVL.
- A CONTEXT may CONTAIN any number of RELEASF's.
- A CONTEXT may CONTAIN at most one SECURITY.
- A CONTEXT may CC.NTAIN any number of TRACK's.
- A CONTEXT may CONTAIN at most one VALSTAT.
- A CONTEXT may CONTAIN any number of VERSION's.
- A CONTEXT may CONTAIN at most one VERSTAT.
- A CONTEXT may BE-CONTAINED-IN any number of COMMON-ELEMENT's.
- A CONTEXT may TEST any number of PRECOND's.

030 DEFAULT - Default Menu Choice or Fill-in-the-Blank Response

Description: Represents a text string which will be used as the initial entry in the fill-in-the-blank form. In case of "MENU", the default is a pointer to one of the possible "CHOICE" responses.

Attribute of: FILLIN, MENU

Entity Type Relationships/Constraints: See Section 50.2.6

- A DEFAULT is an entity type.
- A DEFAULT may be OF any number of FILLIN's.
- A DEFAULT may be OF any number of MENU's.
- A DEFAULT must be WITH exactly one DEFAULT-DATA.

031 DEFAULT-DATA - A Label Type of DEFAULT

Data Format: 80 A L

Label Type Descriptions/Constraints: See Section 50.2.6

- A DEFAULT-DATA is a label type with representation Character 80.
- A DEFAULT-DATA may be OF at most one DEFAULT.

032 DESCINFO - Descriptive Information

Description: Defines general purpose, non-procedural, or narrative information such as theory of operation, schematics, wiring diagrams, etc. It is a very flexible, general purpose information entity.

Attribute of: DESCINFO, OPERINFO, SYSTEM

Entity Type Relationships/Constraints: See Section 50.2.3

- A DESCINFO is an entity type.
- A DESCINFO is a subtype of COMMON-ELEMENT.
- A DESCINFO may ASSERT any number of ASSERTION's.
- A DESCINFO may DISPLAY any number of GRAPHIC's.
- A DESCINFO may DISPLAY at most one TEXT.
- A DESCINFO may INVOKE any number of AUDIO's.
- A DESCINFO may INVOKE any number of PROCESS's.
- A DESCINFO may INVOKE any number of VIDEO's.
- A DESCINFO may be SUB-DESCINFO-OF any number of DESCINFO's.
- A DESCINFO may BE-REFERENCED-BY any number of OPERINFO's.
- A DESCINFO may BE-REFERENCED-BY any number of SYSTEM's.
- A DESCINFO may PRESENT any number of PROMPT's.
- A DESCINFO may PRESENT any number of TABLE's.
- A DESCINFO may REFER-TO any number of ANNOT's.
- A DESCINFO may REFER-TO any number of DESCINFO's.

033 DICTITEM - Dictionary Item

Description: References an element which allows the creation of a dictionary of terms used in the database and which can be employed in a help function later in the presentation system.

Attribute of: NONE

Entity Type Relationships/Constraints: See Section 50.2.10

- A DICTITEM is an entity type.
- A DICTITEM is a subtype of COMMON-ELEMENT.
- A DICTITEM must REFER-TO one or more ELMNTREF's.

034 ELMNTREF - Element Reference

Description: A reference to an element of any type. In some cases, CDM elements may need to refer to or link a mixture of element types.

Attribute of: ATTVALUE, DICTITEM, ENTRY, LIST, PROPERTY, REQCOND, and XREF

Entity Type Relationships/Constraints: See Section 50.2.20

- An ELMNTREF is an entity type.
- An ELMNTREF may BE-REFERENCED-BY any number of DICTITEM's.
- An ELMNTREF may BE-REFERENCED-BY any number of ATTVALUE's.
- An ELMNTREF may BE-REFERENCED-BY any number of ENTRY's.
- An ELMNTREF may BE-REFERENCED-BY any number of LIST's.
- An ELMNTREF may BE-REFERENCED-BY any number of PROPERTY's.
- An ELMNTREF may BE-REFERENCED-BY any number of REQCOND's.
- An ELMNTREF may BE-REFERENCED-BY any number of XREF's.
- An ELMNTREF must be SAME-AS exactly one CDM-ELEMENT.

035 ENTRY - Table entry

Description: References the cells or parts of a table which may take the form of text or any other internal element.

Attribute of: TABLE

Entity Type Relationships/Constraints: See Section 50.2.12

- An ENTRY is an entity type.
- An ENTRY is a subtype of CDM-ELEMENT.
- An ENTRY must CONTAIN exactly one COL.
- An ENTRY must CONTAIN exactly one ROW.
- An ENTRY may DISPLAY at most one TEXT.
- An ENTRY may BE-CONTAINED-IN any number of TABLE's.
- An ENTRY may REFER-TO at most one ELMNTREF.

036 EQUIP - Maintenance Equipment

Description: Identifies the equipment needed to perform a particular task or step. Usually refers to a piece of test equipment.

Attribute of: STEP, TASK

Entity Type Relationships/Constraints: See Section 50.2.5

- An EQUIP is an entity type.
- An EQUIP is a subtype of COMMON-ELEMENT.
- An EQUIP may be ALT-EQUIP-OF any number of EQUIP's.
- An EQUIP must HAVE exactly one QTY.
- An EQUIP may BE-REQUIRED-BY any number of STEP's.
- An EQUIP may BE-REQUIRED-BY any number of TASK's.

037 ESTTIME - Estimated Time for Completion

Description: Represents the amount of time required for the corresponding task/step to be completed.

Attribute of: STEP, TASK

Entity Type Relationships/Constraints: See Section 50.2.4

- An ESTTIME is an entity type.
- An ESTTIME may BE-REQUIRED-BY any number of STEP's.
- An ESTTIME may BE-REQUIRED-BY any number of TASK's.
- An ESTTIME must be WITH exactly one TIME.

038 EXREF-ID - A Label Type of EXREFID

Data Format: 10 X L

Label Type Descriptions/Constraints: See Section 50.2.20

- An EXREF-ID is a label type with representation Alphanumeric 10.
- An EXREF-ID may be OF at most one EXREFID.

039 EXREFID - External Reference

Description: Represents an external reference that is an element

in another file or database, external to the Content Data Model.

Attribute of: AUDIO, PROCESS, VIDEO, XREF

Entity Type Relationships/Constraints: See Section 50.2.20

An EXREFID is an entity type.

An EXREFID may BE-REFERENCED-BY any number of AUDIO's.

An EXREFID may BE-REFERENCED-BY any number of PROCESS's.

An EXREFID may BE-REFERENCED-BY any number of VIDEO's.

An EXREFID may BE-REFERENCED-BY any number of XREF's.

An EXREFID must be WITH exactly one EXREF-ID.

040 FAULT - Fault

Description: References a potential failure which may occur within a system.

Attribute of: FAULTINF, RECT

Entity Type Relationships/Constraints: See Section 50.2.16

A FAULT is an entity type.

A FAULT is a subtype of COMMON-ELEMENT.

A FAULT may DISPLAY at most one TEXT.

A FAULT may be EXCULPATING any number of FLTSTATE's.

A FAULT may be IMPLICATING any number of FLTSTATE's.

If a FAULT is EXCULPATING a FLTSTATE,

then that same FAULT cannot be IMPLICATING that same FLTSTATE.

A FAULT may HAVE at most one MTBF.

A FAULT must IDENTIFY one or more RECT's.

A FAULT may BE-IDENTIFIED-BY any number of FAULTINF's.

A FAULT may POINT-TO any number of PARTINFO's.

041 FAULTINF - Fault Information Data

Description: References the collection of all faults and tests which pertain to a given system.

Attribute of: FAULTINF, SYSTEM

Entity Type Relationships/Constraints: See Section 50.2.15

A FAULTINF is an entity type.

A FAULTINF is a subtype of COMMON-ELEMENT.

A FAULTINF must IDENTIFY one or more FAULT's.

A FAULTINF must IDENTIFY one or more TEST's.

A FAULTINF may BE-REFERENCED-BY any number of SYSTEM's.

042 FILE - Data File

Description: Represents the name of an external file which may contain graphics, audio, video, or software information.

Attribute of: AUDIO, GRPHPRIM, PROCESS, VIDEO

Entity Type Relationships/Constraints: See Section 50.2.14

- A FILE is an entity type.
- A FILE may BE-REFERENCED-BY any number of AUDIO's.
- A FILE may BE-REFERENCED-BY any number of GRPHPRIM's.
- A FILE may BE-REFERENCED-BY any number of PROCESS's.
- A FILE may BE-REFERENCED-BY any number of VIDEO's.
- A FILE must be WITH exactly one FILE-DATA.

043 FILE-DATA - A Label Type of FILE

Data Format: 80 X L

Label Type Descriptions/Constraints: See Section 50.2.14

- A FILE-DATA is a label type with representation Alphanumeric 80.
- A FILE-DATA may be OF at most one FILE.

044 FILLIN - Fill-in-the-blank

Description: References a fill-in-the-blank form which will be displayed to the technician. This element can have a default text string for its initial entry in the fill-in-the-blank form.

Attribute of: PROMPT

Entity Type Relationships/Constraints: See Section 50.2.6

- A FILLIN is an entity type.
- A FILLIN is a subtype of COMMON-ELEMENT.
- A FILLIN must ASSERT exactly one PROPERTY.
- A FILLIN may HAVE at most one DEFAULT.
- A FILLIN must DISPLAY exactly one TEXT.
- A FILLIN may be OF any number of PROMPT's.

045 FLTSTATE - Fault State

Description: References a list of faults (i.e. an ambiguity group) which are implicated in the fault state. It provides information necessary to select the next diagnostic test, which may be done explicitly in a decision tree or by the software in a dynamic model.

Attribute of: FLTSTATE, OUTCOME

Entity Type Relationships/Constraints: See Section 50.2.16

- A FLTSTATE is an entity type.
- A FLTSTATE is a subtype of COMMON-ELEMENT.
- A FLTSTATE may DISPLAY at most one TEXT.
- A FLTSTATE may be EXCULPATED-BY any number of FAULT's.
- A FLTSTATE may be IMPLICATED-BY any number of FAULT's.
- If a FLTSTATE is EXCULPATED-BY a FAULT,
then that same FLTSTATE may not be IMPLICATED by that same FAULT.
- A FLTSTATE may IDENTIFY at most one TEST.
- A FLTSTATE may be IDENTIFIED-BY any number of OUTCOME's.

A FLTSTATE may be SCALED-BY any number of WEIGHT's.

046 FOLLOWON - Follow on Task

Description: References a list of tasks which are required after completion of the current task, but reside elsewhere in the database.

Attribute of: TASK

Entity Type Relationships/Constraints: See Section 50.2.4

A FOLLOWON is an entity type.

A FOLLOWON may be REFERENCED-BY any number of TASK's.

A FOLLOWON must be WITH exactly one REFID.

047 GOVSTD - Government Standard

Description: A document that establishes engineering and technical requirements for processes, procedures, practices and methods that has been adopted as a standard. It also establishes requirements for selection, application and design criteria for materials.

Attribute of: CONSUM

Entity Type Relationships/Constraints: See Section 50.2.11

A GOVSTD is an entity type.

A GOVSTD may SPECIFY any number of CONSUM's.

A GOVSTD must be WITH exactly one GOVSTD-CODE.

048 GOVSTD-CODE - A Label Type of GOVSTD

Data Format: 15 X L

Label Type Descriptions/Constraints: See Section 50.2.11

A GOVSTD-CODE is a label type with representation Alphanumeric 15.

A GOVSTD-CODE may be OF at most one GOVSTD.

049 GRAPHIC - Graphic Element

Description: References an element used to group graphic primitives into a composite graphic. It also supports those transformations needed to be performed on the graphic primitives to scale, rotate, translate, etc. the individual primitives into a composite image.

Attribute of: DESCINFO, PARTINFO, STEP

Entity Type Relationships/Constraints: See Section 50.2.13

A GRAPHIC is an entity type.

A GRAPHIC is a subtype of COMMON-ELEMENT.

A GRAPHIC may DISPLAY at most one TEXT.

A GRAPHIC must INVOKE one or more GRPHPRIM's.

A GRAPHIC may be DISPLAYED-BY any number of DESCINFO's.

A GRAPHIC may be DISPLAYED-BY any number of STEP's.

- A GRAPHIC may be FOCUSED BY any number of GRAPGIC's.
- A GRAPHIC may be IDENTIFIED-BY any number of PARTINFO's.
- A GRAPHIC may SPECIFY any number of MINSIZE's.
- A GRAPHIC may SPECIFY at most one PENPATT.
- A GRAPHIC may SPECIFY at most one PENSHAPE.
- A GRAPHIC may SPECIFY any number of TRANSFRM's.
- A GRAPHIC may SPECIFY any number of WINDOW's.
- A GRAPHIC may SPECIFY FOCUS any number of GRAPHIC's.

050 GRPHPRIM - Graphic Primitive

Description: References a single graphic component, which when combined with other primitives, can become a composite graphic. A graphic primitive references a file which contains the detailed graphic information in the form of CGM, IGES, FAX, or DXF graphic codes.

Attribute of: GRAPHIC

Entity Type Relationships/Constraints: See Section 50.2.14

- A GRPHPRIM is an entity type.
- A GRPHPRIM is a subtype of COMMON-ELEMENT.
- A GRPHPRIM may CONTAIN at most one PCDATA.
- A GRPHPRIM may DISPLAY at most one TEXT.
- A GRPHPRIM may be INVOKED-BY any number of GRAPHIC's.
- A GRPHPRIM may be INVOKED-BY any number of FOCUS's.
- A GRPHPRIM may REFER-TO at most one FILE.
- A GRPHPRIM may SPECIFY at most one CODING.
- A GRPHPRIM may SPECIFY any number of MINSIZE's.
- A GRPHPRIM may SPECIFY at most one PENPATT.
- A GRPHPRIM may SPECIFY at most one PENSHAPE.
- A GRPHPRIM may SPECIFY at most one START.
- A GRPHPRIM may SPECIFY at most one STOP.
- A GRPHPRIM may SPECIFY any number of TRANSFRM's.
- A GRPHPRIM may SPECIFY any number of WINDOW's.

051 HCI - Hardness Critical Index

Description: Indicates that a particular piece of equipment has nuclear survivability requirements which cannot be degraded during maintenance and operation.

Attribute of: PARTBASE

Entity Type Relationships/Constraints: See Section 50.2.18

- A HCI is an entity type.
- A HCI may be OF any number of PARTBASE's.
- A HCI must be WITH exactly one HCI-CODE.

052 HCI-CODE - A Label Type of HCI

Data Format: 1 X F

Label Type Descriptions/Constraints: See Section 50.2.18
A HCI-CODE is a label type with representation Character 1.
A HCI-CODE may be OF at most one HCI.

053 ID - A Label Type of CDM-ELEMENT

Data Format: 15 X L

Description: Designates a unique machine generated identifier for an element.

Attribute of: CDM-ELEMENT

Label Type Descriptions/Constraints: See Section 50.2.1
An ID is a label type with representation Character 15.
An ID may be OF at most one CDM-ELEMENT.

054 INDX-NUMBER - A Label Type of INDXNUM

Data Format: 15 X L

Label Type Descriptions/Constraints: See Section 50.2.17
An INDX-NUMBER is a label type with representation Alphanumeric 15.
An INDX-NUMBER may be OF at most one INDXNUM.

055 INDXNUM - Index Number

Description: References the maintainers' view of part information. It describes an item in term of its 'refdes' which categorizes parts by their place in the system-subsystem hierarchy.

Attribute of: PARTINFO

Entity Type Relationships/Constraints: See Section 50.2.17
An INDXNUM is an entity type.
An INDXNUM may be REFERRED by any number of PARTINFO's.
An INDXNUM must be WITH exactly one INDX-NUMBER.

056 ITEMID - A Label Type of COMMON-ELEMENT

Data Format: 32 X L

Description: Represents the specific equipment item which is associated with the element it pertains to. It could be a refdes, ssn #, part #, etc. depending on which element it refers to.

Attribute of: COMMON-ELEMENT, PARTBASE

Label Type Descriptions/Constraints: See Section 50.2.2
An ITEMID is a label type with representation Alphanumeric 32.
An ITEMID may be OF any number of COMMON-ELEMENT's.

An ITEMID may be OF any number of PARTBASE's.

Example(s):

task/itemid: "94-50-03"

057 LIST - List Element

Description: References a general purpose element which can be referenced from any element to show a grouping of elements with a common context.

Attribute of: NONE

Entity Type Relationships/Constraints: See Section 50.2.20

A LIST is an entity type.

A LIST is a subtype of CDM-ELEMENT.

A LIST must CONTAIN exactly one CONTEXT.

A LIST must REFER-TO one or more ELMNTREF's.

A LIST may be WITH at most one LIST-NAME.

A LIST must be WITH exactly one REFID.

058 LIST-NAME - A Label Type of LIST

Data Format: 15 X L

Label Type Descriptions/Constraints: See Section 50.2.20

A LIST-NAME is a label type with representation Alphanumeric 15.

A LIST-NAME may be OF any number of LIST's.

059 MAINT-LEVEL - A Label Type of MAINTLVL

Data Format: 10 X L

Label Type Descriptions/Constraints: See Section 50.2.19

A MAINT-LEVEL is a label type with representation Alphanumeric 10.

A MAINT-LEVEL may be OF at most one MAINTLVL.

Example(s):

context/maintlvl: "olevel", or "ilevel"

060 MAINTLVL - Maintenance Level

Description: Represents the appropriate maintenance level for repair of an item and helps provide the correct information for that level. Valid levels are: organizational, intermediate, and depot.

Attribute of: CONTEXT

Entity Type Relationships/Constraints: See Section 50.2.19

A MAINTLVL is an entity type.

A MAINTLVL may be OF any number of CONTEXT's.

A MAINTLVL must be WITH exactly one MAINT-LEVEL.

061 MENU - Menu of Choices

Description: References a type of prompt which is a menu choice. A menu choice consists of a question (text), a property, and a set of possible responses (choices). The user chooses a response, the value associated with the choice selected is asserted for the value of the property associated with the menu.

Attribute of: PROMPT

Entity Type Relationships/Constraints: See Section 50.2.6

- A MENU is an entity type.
- A MENU is a subtype of COMMON-ELEMENT.
- A MENU must ASSERT exactly one PROPERTY.
- A MENU may be CALLED-BY any number of PROMPT's.
- A MENU must DISPLAY exactly one TEXT.
- A MENU must HAVE one or more CHOICE's.
- A MENU may HAVE any number of DEFAULT's.
- A MENU may HAVE at most one SELECT.

062 MFG-CODE - A Label Type of MFGCODE

Data Format: 10 X L

Label Type Descriptions/Constraints: See Section 50.2.19

- A MFG-CODE is a label type with representation Alphanumeric 10.
- A MFG-CODE may be OF at most one MFGCODE.

063 MFGCODE - Manufacturer's Code.

Description: Represents the code a manufacturer uses in-house to represent parts.

Attribute of: CONSUM

Entity Type Relationships/Constraints: See Section 50.2.19

- A MFGCODE is an entity type.
- A MFGCODE may SPECIFY any number of CONSUM's.
- A MFGCODE must be WITH exactly one MFG-CODE.

064 MILSPEC - Military Specification Number.

Description: Represents the exact specification for each item bought by the government.

Attribute of: CONSUM

Entity Type Relationships/Constraints: See Section 50.2.19

- A MILSPEC is an entity type.
- A MILSPEC may SPECIFY any number of CONSUM's.

A MILSPEC must be WITH exactly one MILSPEC-CODE.

065 MILSPEC-CODE - A Label Type of MILSPEC

Data Format: 10 X L

Label Type Descriptions/Constraints: See Section 50.2.19

A MILSPEC-CODE is a label type with representation Alphanumeric 10.

A MILSPEC-CODE may be OF at most one MILSPEC.

066 MINSIZE - Minimum Size

Description: Represents the minimum size required to display a graphic to satisfy the technicians' requirements. The size of the image should be provided in the minimum number of minutes of arc, which must be subtended by the image on the user's eye to make the graphic readable to the user.

Attribute of: GRAPHIC, GRPHPRIM

Entity Type Relationships/Constraints: See Section 50.2.13

A MINSIZE is an entity type.

A MINSIZE may be SPECIFIED-BY any number of GRPHPRIM's.

A MINSIZE may be SPECIFIED-BY any number of GRAPHIC's.

A MINSIZE must be WITH exactly one MINSIZE-CODE.

A MINSIZE-CODE may be OF at most one MINSIZE.

067 MINSIZE-CODE - A Label Type of MINSIZE

Data Format: 10 N R

Label Type Descriptions/Constraints: See Section 50.2.13

A MINSIZE-CODE is a label type with representation Numeric 10.

A MINSIZE-CODE be OF at most one MINSIZE.

068 MTBF - Mean Time Between Failure

Description: Represents the average time a specific part or system should work between failures.

Attribute of: FAULT, PARTINFO

Entity Type Relationships/Constraints: See Section 50.2.17

A MTBF is an entity type.

A MTBF may be OF any number of FAULT's.

A MTBF may be OF any number of PARTINFO's.

A MTBF must be WITH exactly one MTBF-TIME.

069 MTBF-TIME - A Label Type of MTBF

Data Format: 10 N R

Label Type Descriptions/Constraints: See Section 50.2.17
A MTBF-TIME is a label type with representation Numeric 10.
A MTBF-TIME may be OF at most one MTBF.

070 NAME - A Label Type of COMMON-ELEMENT

Data Format: 80 X L

Description: Represents the textual name or title of an element.

Attribute of: COMMON-ELEMENT, PARTBASE

Label Type Descriptions/Constraints: See Section 50.2.2
A NAME is a label type with representation Alphanumeric 80.
A NAME may be OF any number of COMMON-ELEMENT's.
A NAME may be OF at most one PARTBASE.

071 NOTE - Note

Description: References additional information to aid the technician in completing a given step. A NOTE is used in technical information to emphasize an especially important procedure or condition.

Attribute of: STEP, TASK

Entity Type Relationships/Constraints: See Section 50.2.5
A NOTE is an entity type.
A NOTE is a subtype of COMMON-ELEMENT.
A NOTE must DISPLAY exactly one TEXT.
A NOTE may be REFERENCED-BY any number of STEP's.
A NOTE may be REFERENCED-BY any number of TASK's.

072 NOUNID - A Label Type of PARTINFO

Data Format: 80 X L

Description: Represents a general name of a part.

Attribute of: PARTINFO

Label Type Descriptions/Constraints: See Section 50.2.17
A NOUNID is a label type with representation Alphanumeric 80.
A NOUNID may be OF any number of PARTINFO's.

073 NOUNTYPE - A Label Type of PARTINFO

Data Format: 80 X L

Description: Represents more specific descriptors which differentiate part names.

Attribute of: PARTINFO

Label Type Descriptions/Constraints: See Section 50.2.17

A NOUNTYPE is a label type with representation Alphanumeric 80.

A NOUNTYPE may be OF any number of PARTINFO's.

074 NSN - A Label Type of PARTBASE

Data Format: 20 X L

Description: The NSN is assigned under the Federal Cataloguing Program and/or NATO codification of equipment system to each approved item identification to provide an unique identification of an item of supply within a specified Federal Supply Classification (FSC). The field consists of a three-character prefix, a thirteen-characters National Stock Number (NSN), and a four-character suffix code. For applicable, see DoD 4100.38-M.

Attribute of: PARTBASE

Label Type Descriptions/Constraints: See Section 50.2.18

A NSN is a label type with representation Alphanumeric 20.

A NSN may be OF at most one PARTBASE.

075 NUMBER - A Label Type of QTY

Data Format: 10 N R

Label Type Descriptions/Constraints: See Section 50.2.4

A NUMBER is a label type with representation Numeric 10.

A NUMBER may be OF at most one QTY.

076 OP - Operator

Description: Represents the operator within a comparison between a property and a value.

Attribute of: PRECOND

Entity Type Relationships/Constraints: See Section 50.2.7

An OP is an entity type.

An OP may be OF any number of PRECOND's.

An OP must be WITH exactly one OP-DATA.

077 OP-DATA - A Label Type of OP

Data Format: 3 A L

Label Type Descriptions/Constraints: See Section 50.2.7

An OP-DATA is a label type with representation Character 3.

An OP-DATA may be OF at most one OP.

Example(s): "eq", "lt", "lte", "gt", "gte", or "in".

078 OPERINFO - Operational information

Description: References procedural task information and descriptive information required for operating the system in question.

Attribute of: SYSTEM

Entity Type Relationships/Constraints: See Section 50.2.3

An OPERINFO is an entity type.

An OPERINFO is a subtype of COMMON-ELEMENT.

An OPERINFO may be REFERENCED-BY any number of SYSTEM's.

An OPERINFO may REFER-TO any number of DESCINFO's.

An OPERINFO may REFER-TO any number of TASK's.

079 OUTCOME - Test outcome

Description: References the implicated and exculpated fault states known to be true following a specific test outcome.

Attribute of: TEST

Entity Type Relationships/Constraints: See Section 50.2.15

An OUTCOME is an entity type.

An OUTCOME is a subtype of COMMON-ELEMENT.

An OUTCOME may DISPLAY at most one TEXT.

An OUTCOME must IDENTIFY one or more FLTSTATE's.

An OUTCOME may be PRODUCED-BY any number of TEST's.

An OUTCOME must TEST one or more PRECOND's.

080 PARTBASE - Part Base

Description: References the information about a part that does not change with respect to the place it is used on the vehicle. It represents the supply systems view of part information.

Attribute of: PARTINFO

Entity Type Relationships/Constraints: See Section 50.2.18

A PARTBASE is an entity type.

A PARTBASE is a subtype of CDM-ELEMENT.

A PARTBASE must CONTAIN exactly one HCI.

A PARTBASE must CONTAIN exactly one SMR.

A PARTBASE may CONTAIN at most one TYPE.

A PARTBASE may be IDENTIFIED-BY any number of PARTINFO's.

A PARTBASE must MADE-BY exactly one CAGE.

A PARTBASE may REFER any number of XREF's.

A PARTBASE may be WITH at most one ITEMID.

A PARTBASE may be WITH at most one NAME.

A PARTBASE must be WITH exactly one NSN.

A PARTBASE must be WITH exactly one PARTNUM.

A PARTBASE must be WITH exactly one REFID.

081 PARTINFO - Part Information

Description: References a part in terms of its reference designator, which categorizes parts by their place in the system-subsystem hierarchy. It represents the maintainers' view of part information.

Attribute of: FAULT, SYSTEM

Entity Type Relationships/Constraints: See Section 50.2.17

- A PARTINFO is an entity type.
- A PARTINFO is a subtype of COMMON-ELEMENT.
- A PARTINFO must HAVE exactly one MTBF.
- A PARTINFO must IDENTIFY one or more GRAPHIC's.
- A PARTINFO must IDENTIFY one or more PARTBASE's.
- A PARTINFO may IDENTIFY at most one REPLVL.
- A PARTINFO may be IDENTIFIED-BY any number of SYSTEM's.
- A PARTINFO must BE-IDENTIFIED-BY exactly one USABLON.
- A PARTINFO may be POINTED-TO-BY any number of FAULT's.
- A PARTINFO must BE-REFERENCED-BY exactly one INDXNUM.
- A PARTINFO must REQUIRE exactly one UNITSPER.
- A PARTINFO may be WITH at most one NOUNID.
- A PARTINFO may be WITH at most one NOUNTYPE.
- A PARTINFO must be WITH exactly one REFDES.

082 PARTNUM - Part Number

Data Format: 32 X L

Description: Represents the part number assigned to an item by the supply system.

Attribute of: PARTBASE

Label Type Descriptions/Constraints: See Section 50.2.18

- A PARTNUM is a label type with representation Alphanumeric 32.
- A PARTNUM may be OF any number of PARTBASE's.

083 PCDATA - Text Unit

Description: A text string of parsable character data.

Attribute of: GRPHPRIM, TEXT

Entity Type Relationships/Constraints: See Section 50.2.10

- A PCDATA is an entity type.
- A PCDATA may be CONTAINED-IN any number of GRPHPRIM's.
- A PCDATA may be CONTAINED-IN any number of TEXT's.
- A PCDATA must be WITH exactly one PCTEXT.

084 PCTEXT - A Label Type of PCDATA

Data Format: NARRATIVE

Label Type Descriptions/Constraints: See Section 50.2.10

A PCTEXT is a label type with representation Text 9999.

A PCTEXT may be OF at most one PCDATA.

085 PENPATT - Pen Pattern

Description: Represents the bit map pattern to be used as the pen for drawing lines, points, etc. for a particular graphic.

Attribute of: GRAPHIC, GRPHPRIM

Entity Type Relationships/Constraints: See Section 50.2.13

A PENPATT is an entity type.

A PENPATT may be SPECIFIED-BY any number of GRPHPRIM's.

A PENPATT may be SPECIFIES-BY any number of GRAPHIC's.

A PENPATT must be WITH exactly one PENPATT-DATA.

086 PENPATT-DATA - A Label Type of PENPATT

Data Format: 10 X L

Label Type Descriptions/Constraints: See Section 50.2.13

A PENPATT-DATA is a label type with representation Alphanumeric 10.

A PENPATT-DATA may be OF at most one PENPATT.

087 PENSHAPE - Pen shape

Description: Represents the boundary shape for the pen for drawing lines, points, etc. for a particular graphic.

Attribute of: GRAPHIC, GRPHPRIM

Entity Type Relationships/Constraints: See Section 50.2.13

A PENSHAPE is an entity type.

A PENSHAPE may be SPECIFIED-BY any number of GRAPHIC's.

A PENSHAPE may be SPECIFIED-BY any number of GRPHPRIM's.

A PENSHAPE must be WITH exactly one PENSHAPE-DATA.

088 PENSHAPE-DATA - A Label Type of PENSHAPE

Data Format: 10 X L

Label Type Descriptions/Constraints: See Section 50.2.13

A PENSHAPE-DATA is a label type with representation Alphanumeric 10.

A PENSHAPE-DATA may be OF at most one PENSHAPE.

089 PERSON - Personnel Required

Description: References the number and type of people required to complete a specific step or procedure.

Attribute of: STEP, TASK

Entity Type Relationships/Constraints: See Section 50.2.4

- A PERSON is an entity type.
- A PERSON is a subtype of COMMON-ELEMENT.
- A PERSON must HAVE exactly one QTY.
- A PERSON may be REQUIRED-BY any number of STEP's.
- A PERSON may be REQUIRED-BY any number of TASK's.

090 POLARITY - Logical Polarity

Description: Represents the logical polarity (positive or negative) of a property-value pair for an assertion or precondition. A negative polarity is equivalent to saying that the property-value pair is not true for the assertion or precondition.

Attribute of: PRECOND

Entity Type Relationships/Constraints: See Section 50.2.7

- A POLARITY is an entity type.
- A POLARITY may be OF any number of PRECOND's.
- A POLARITY must be WITH exactly one POLARITY-DATA.

091 POLARITY-DATA - A Label Type of POLARITY

Data Format: 3 A L

Label Type Descriptions/Constraints: See Section 50.2.7

- A POLARITY-DATA is a label type with representation Character 3.
- A POLARITY-DATA may be OF at most one POLARITY.

Example(s): "pos" or "neg".

092 PRECOND - Precondition

Description: States a property-operation-value condition or relation which must be true for its associated data to be applicable to the current context, troubleshooting outcome, or required condition.

Attribute of: CONTEXT, OUTCOME, REQCOND

Entity Type Relationships/Constraints: See Section 50.2.7

- A PRECOND is an entity type.
- A PRECOND is a subtype of CDM-ELEMENT.
- A PRECOND may CONTAIN at most one OP.
- A PRECOND may CONTAIN at most one POLARITY.
- A PRECOND may be TESTED-BY any number of CONTEXT's.
- A PRECOND may be TESTED-BY any number of OUTCOME's.
- A PRECOND may be TESTED-BY any number of REQCOND's.
- A PRECOND must TEST exactly one PROPERTY.

A PRECOND must TEST one or more VALUE's.

093 PROCESS - Software Process

Description: References an external software process (a file, a 1553 instruction, a software program, etc.).

Attribute of: DESCINFO, STEP

Entity Type Relationships/Constraints: See Section 50.2.9

A PROCESS is an entity type.

A PROCESS is a subtype of COMMON-ELEMENT.

A PROCESS may be INVOKED-BY any number of DESCINFO's.

A PROCESS may be INVOKED-BY any number of STEP's.

A PROCESS may REFER-TO at most one EXREFID.

A PROCESS may REFER-TO at most one FILE.

094 PROMPT - User Input Prompt

Description: References a fill-in or menu question, which the system cannot assert itself, to the user. Each prompt is associated with a 'property' which specifies the property which will be asserted, along with the user's response when the prompt is answered.

Attribute of: DESCINFO, STEP

Entity Type Relationships/Constraints: See Section 50.2.6

A PROMPT is an entity type.

A PROMPT is a subtype of COMMON-ELEMENT.

A PROMPT may CALL any number of MENU's.

A PROMPT may CONTAIN any number of FILLIN's.

A PROMPT may DISPLAY at most one TEXT.

A PROMPT may be PRESENTED-BY any number of DESCINFO's.

A PROMPT may be PRESENTED-BY any number of STEP's.

095 PROPERTY - Property

Description: References any text string which defines a property. At runtime, property-value pairs may be asserted or tested by the software.

Attribute of: ASSERTION, FILLIN, MENU, PRECOND

Entity Type Relationships/Constraints: See Section 50.2.8

A PROPERTY is an entity type.

A PROPERTY is a subtype of CDM-ELEMENT.

A PROPERTY must DISPLAY exactly one TEXT.

A PROPERTY may be ASSERTED-BY any number of ASSERTION's.

A PROPERTY may be ASSERTED-BY any number of FILLIN's.

A PROPERTY may be ASSERTED-BY any number of MENU's.

A PROPERTY may be TESTED-BY any number of PRECOND's.

A PROPERTY may REFER-TO at most one ELMNTREF.

096 QTY - Quantity

Description: Specifies the amount of the appropriate consumable required for the associated task/step.

Attribute of: COMSUM, EQUIP, PERSON

Entity Type Relationships/Constraints: See Section 50.2.4

- A QTY is an entity type.
- A QTY may be OF any number of CONSUM's.
- A QTY may be OF any number of EQUIP's.
- A QTY may be OF any number of PERSON's.
- A QTY must be WITH exactly one NUMBER.

097 RANGE - Range of Values

Description: Specifies the boundaries for valid choices or outcomes according to the element containing the range.

Attribute of: FILLIN, TEST

Entity Type Relationships/Constraints: See Section 50.2.15

- A RANGE is an entity type.
- A RANGE may be OF any number of FILLIN's.
- A RANGE may REPRESENT any number of TEST's.
- A RANGE must be WITH exactly one RANGE-DATA.

098 RANGE-DATA - A Label Type of RANGE

Data Format: 80 X L

Label Type Descriptions/Constraints: See Section 50.2.15

- A RANGE-DATA is a label type with representation Alphanumeric 80.
- A RANGE-DATA may be OF at most one RANGE.

099 RECT - Rectification

Description: References the repair procedures necessary to fix an associated fault. They also point to tests, which are usually checkout tasks to verify that the rectification was successful. Also, it has a fault attribute used to identify all the faults repaired by the rectification.

Attribute of: FAULT

Entity Type Relationships/Constraints: See Section 50.2.16

- A RECT is an entity type.
- A RECT is a subtype of COMMON-ELEMENT.
- A RECT may DISPLAY at most one TEXT.
- A RECT may IDENTIFY at most one ACTION.

A RECT must IDENTIFY one or more TASK's.
A RECT may IDENTIFY any number of TEST's.
A RECT may be IDENTIFIED-BY any number of FAULT's.
A RECT may be PERFORMED-BY at most one AGENT.

100 REFDES - Reference Designator

Data Format: 10 X L

Description: Specifies an identifier given to a part to show its place in the system architecture the maintainer sees.

Attribute of: PARTINFO

Label Type Descriptions/Constraints: See Section 50.2.17

A REFDES is a label type with representation Alphanumeric 10.

A REFDES may be OF at most one PARTINFO.

101 REFID - A Label Type of COMMON-ELEMENT

Data Format: 10 X L

Description: Designates a non-unique reference identifier for an element. If there are a number of elements which can be used interchangeably (the particular element to be used is selected by "CONTEXT"), then all of these interchangeable elements would have the same REFID.

Attribute of: COMMON-ELEMENT, LIST, PARTBASE

Label Type Descriptions/Constraints: See Section 50.2.2

A REFID is a label type with representation Alphanumeric 10.

A REFID may be OF any number of COMMON-ELEMENT's.

A REFID may be OF at most one FOLLOWON.

A REFID may be OF any number of LIST's.

A REFID may be OF any number of PARTBASE's.

102 RELATION - Related to Cross Reference

Description: Specifies a text string used to describe the nature of the cross reference (theory, schematic, etc.).

Attribute of: XREF

Entity Type Relationships/Constraints: See Section 50.2.20

A RELATION is an entity type.

A RELATION may be CONTAINED-IN any number of XREF's.

A RELATION must be WITH exactly one RELATION-DATA.

103 RELATION-DATA - A Label Type of RELATION

Data Format: 10 X L

Label Type Descriptions/Constraints: See Section 50.2.20

A RELATION-DATA is a label type with representation Alphanumeric 10.

A RELATION-DATA may be OF at most one RELATION.

104 RELEASE - Release Specification

Description: Specifies in which countries the document may be released.

Attribute of: CONTEXT

Entity Type Relationships/Constraints: See Section 50.2.19

A RELEASE is an entity type.

A RELEASE may be OF any number of CONTEXT's.

A RELEASE must be WITH exactly one RELEASE-CODE.

105 RELEASE-CODE - A Label Type of RELEASE

Data Format: 10 X L

Label Type Descriptions/Constraints: See Section 50.2.19

A RELEASE-CODE is a label type with representation Alphanumeric 10.

A RELEASE-CODE may be OF at most one RELEASE.

106 REP-LEVEL - A Label Type of REPLVL

Data Format: 10 X L

Label Type Descriptions/Constraints: See Section 50.2.17

A REP-LEVEL is a label type with representation Alphanumeric 10.

A REP-LEVEL may be OF at most one REPLVL.

107 REPLVL - Replenishment Level

Description: Represents the minimum quantity of a part in stock, which will trigger a reorder or stock action.

Attribute of: PARTINFO

Entity Type Relationships/Constraints: See Section 50.2.17

A REPLVL is an entity type.

A REPLVL may be IDENTIFIED-BY any number of PARTINFO's.

A REPLVL must be WITH exactly one REP-LEVEL.

108 REQCOND - Required Condition

Description: References a list of preliminary conditions which must be satisfied before performing a step or step sequence.

Attribute of: STEP, TASK

Entity Type Relationships/Constraints: See Section 50.2.4

A REQCOND is an entity type.

A REQCOND is a subtype of COMMON-ELEMENT.

A REQCOND may be TESTED-BY any number of STEP's.

A REQCOND may be TESTED-BY any number of TASK's.

A REQCOND may REFER-TO any number of ELMNTREF's.

A REQCOND may TEST any number of PRECOND's.

109 ROW - Row of a Table

Description: Represents a particular row in a table.

Attribute of: ENTRY

Entity Type Relationships/Constraints: See Section 50.2.12

A ROW is an entity type.

A ROW may be CONTAINED-IN any number of ENTRY's.

A ROW must be WITH exactly one ROW-NUMBER.

110 ROW-NUMBER - A Label Type of ROW

Data Format: 10 N R

Label Type Descriptions/Constraints: See Section 50.2.12

A ROW-NUMBER is a label type with representation Numeric 10.

A ROW-NUMBER may be OF at most one ROW.

111 SECURITY - Security Classification

Description: Specifies the security level of the information.

Attribute of: CONTEXT

Entity Type Relationships/Constraints: See Section 50.2.19

A SECURITY is an entity type.

A SECURITY may be OF any number of CONTEXT's.

A SECURITY must be WITH exactly one SECURITY-CODE.

112 SECURITY-CODE - A Label Type of SECURITY

Data Format: 10 A L

Label Type Descriptions/Constraints: See Section 50.2.19

A SECURITY-CODE is a label type with representation Character 10.

A SECURITY-CODE may be OF at most one SECURITY.

Example(s): "uc", "c", "s", or "ts".

113 SELECT - Number of Selections on a menu

Description: Represents either a single selection or multiple selection of choices from a menu element.

Attribute of: MENU

Entity Type Relationships/Constraints: See Section 50.2.6

A SELECT is an entity type.

A SELECT may be OF any number of MENU's.

A SELECT must be WITH exactly one SELECT-DATA.

114 SELECT-DATA - A Label Type of SELECT

Data Format: 1 A F

Label Type Descriptions/Constraints: See Section 50.2.6

A SELECT-DATA is a label type with representation Character 1.

A SELECT-DATA may be OF at most one SELECT.

Example(s): "single" or "multiple"

115 SMR - Source Maintenance Reliability Code

Description: Represents the code used to identify how recoverable a part is (recyclable, o-level, i-level, replace, etc.).

Attribute of: PARTBASE

Entity Type Relationships/Constraints: See Section 50.2.18

A SMR is an entity type.

A SMR may be OF any number of PARTBASE's.

A SMR must be WITH exactly one SMR-CODE.

116 SMR-CODE - A Label Type of SMR

Data Format: 6 X L

Label Type Descriptions/Constraints: See Section 50.2.18

A SMR-CODE is a label type with representation Alphanumeric 6.

A SMR-CODE may be OF at most one SMR.

117 START - Starting Point in a Graphics File

Description: Identifies a location within a graphic file where the presentation system should start reading graphics codes for the specific graphic being presented.

Attribute of: GRPHPRIM

Entity Type Relationships/Constraints: See Section 50.2.14

A START is an entity type.

A START may be SPECIFIED-BY any number of GRPHPRIM's.

A START must be WITH exactly one STOP-DATA.

118 START-DATA - A Label Type of START

Data Format: 15 X L

Label Type Descriptions/Constraints: See Section 50.2.14

A START-DATA is a label type with representation Alphanumeric 15.

A START-DATA may be OF at most one START.

119 STEP - Task Step

Description: References a procedural step within a task.

Attribute of: STEP, TASK

Entity Type Relationships/Constraints: See Section 50.2.5

A STEP is an entity type.

A STEP is a subtype of COMMON-ELEMENT.

A STEP may ASSERT any number of ASSERTION's.

A STEP may CONTAIN any number of ANNOT's.

A STEP may DISPLAY any number of GRAPHIC's.

A STEP must DISPLAY exactly one TEXT.

A STEP may INVOKE any number of AUDIO's.

A STEP may INVOKE any number of PROCESS's.

A STEP may INVOKE any number of VIDEO's.

A STEP may be SUB-STEP-OF any number of STEP's.

A STEP may be REFERENCED-BY any number of TASK's.

A STEP may PRESENT any number of PROMPT's.

A STEP may PRESENT any number of TABLE's.

A STEP may REFER-TO any number of CAUTION's.

A STEP may REFER-TO any number of NOTE's.

A STEP may be SUB-STEP-OF any number of STEP's.

A STEP may REFER-TO any number of WARNING's.

A STEP may REQUIRE any number of CONSUM's.

A STEP may REQUIRE any number of EQUIP's.

A STEP may REQUIRE at most one ESTTIME.

A STEP may REQUIRE any number of PERSON's.

A STEP may TEST any number of REQCOND's.

A STEP may USE any number of VERB's.

120 STOP - Stopping Point in a Graphics File

Description: Identifies a location within a graphic file where the presentation system should stop reading graphics codes for the specific graphic being presented.

Attribute of: GRPHPRIM

Entity Type Relationships/Constraints: See Section 50.2.14

A STOP is an entity type.

A STOP may be SPECIFIED-BY any number of GRPHPRIM's.

A STOP must be WITH exactly one STOP-DATA.

121 STOP-DATA - A Label Type of STOP

Data Format: 15 X L

Label Type Descriptions/Constraints: See Section 50.2.14

A STOP-DATA is a label type with representation Alphanumeric 15.

A STOP-DATA may be OF at most one START.

A STOP-DATA may be OF at most one STOP.

122 SYSTEM - System

Description: References a vehicle, system, subsystem, or subassembly element in the equipment hierarchy. A SYSTEM may reference procedural task information, descriptive information, parts information, fault information, or operational information.

Attribute of: SYSTEM

Entity Type Relationships/Constraints: See Section 50.2.3

A SYSTEM is an entity type.

A SYSTEM is a subtype of COMMON-ELEMENT.

A SYSTEM may IDENTIFY any number of PARTINFO's.

A SYSTEM may be SUB-SYSTEM-OF any number of SYSTEM's.

A SYSTEM may REFER-TO any number of DESCINFO's.

A SYSTEM may REFER-TO any number of FAULTINF's.

A SYSTEM may REFER-TO any number of OPERINFO's.

A SYSTEM may REFER-TO any number of SYSTEM's.

A SYSTEM may REFER-TO any number of TASK's.

123 TABLE - Tabular Information

Description: References the components required to construct a table or chart.

Attribute of: DESCINFO, STEP

Entity Type Relationships/Constraints: See Section 50.2.12

A TABLE is an entity type.

A TABLE is a subtype of COMMON-ELEMENT.

A TABLE must CONTAIN one or more COLHDDEF's.

A TABLE must CONTAIN one or more ENTRY's.

A TABLE may be PRESENTED-BY any number of DESCINFO's.

A TABLE may be PRESENTED-BY any number of STEP's.

124 TASK - Task

Description: Specifies a task as a set of directive steps which make up a specific maintenance procedure. A maintenance procedure could be a preventive or corrective maintenance task. A procedural task is composed of STEPs, and connects all text, graphics, messages, prompts, and references required to convey the step together.

Attribute of: OPERINFO, RECT, SYSTEM, TEST

Entity Type Relationships/Constraints: See Section 50.2.4

- A TASK is an entity type.
- A TASK is a subtype of COMMON-ELEMENT.
- A TASK may be IDENTIFIED any number of RECT's.
- A TASK may be IDENTIFIED-BY any number of TEST's.
- A TASK may be REFERENCED-BY any number of OPERINFO's.
- A TASK may be REFERENCED-BY any number of SYSTEM's.
- A TASK may REFER-TO any number of CAUTION's.
- A TASK may REFER-TO any number of FOLLOWON's.
- A TASK may REFER-TO any number of NOTE's.
- A TASK must REFER-TO one or more STEP's.
- A TASK may REFER-TO any number of WARNING's.
- A TASK may REQUIRE any number of CONSUM's.
- A TASK may REQUIRE any number of EQUIP's.
- A TASK may REQUIRE at most one ESTTIME.
- A TASK may REQUIRE any number of PERSON's.
- A TASK may TEST any number of REQCOND's.
- A TASK may USE any number of VERB's.

125 TECHINFO - Technical Information

Description: References all elements. It is the top level of the CDM.

Attribute of: CDM-ELEMENT

Entity Type Relationships/Constraints: See Section 50.2.3

- A TECHINFO is an entity type.
- A TECHINFO is a subtype of COMMON-ELEMENT.
- A TECHINFO may CONTAIN any number of SYSTEM's.

126 TEST - Fault Isolation Test

Description: References a diagnostic test used to guide the technician toward a rectification during troubleshooting.

Attribute of: FAULTINF, FLTSTATE, RECT

Entity Type Relationships/Constraints: See Section 50.2.15

- A TEST is an entity type.
- A TEST is a subtype of COMMON-ELEMENT.
- A TEST may DISPLAY at most one TEXT.
- A TEST must IDENTIFY one or more TASK's.
- A TEST may be IDENTIFIED-BY any number of FAULTINF's.
- A TEST may be IDENTIFIED-BY any number of FLTSTATE's.
- A TEST may be IDENTIFIED-BY any number of RECT's.
- A TEST may be PERFORMED-BY at most one AGENT.
- A TEST may be REPRESENTED-BY at most one RANGE.
- A TEST must PRODUCE one or more OUTCOME's.

127 TEXT - Text

Description: Represents a text string of parsable character data.
This can be a letter, number, word, sentence, paragraph, etc.

Attribute of: ANNOT, CAUTION, CHOICE, DESCINFO, ENTRY, FAULT,
FILLIN, FLTSTATE, GRAPHIC, GRPHPRIM, MENU, NOTE, OUTCOME, PROMPT,
PROPERTY, RECT, STEP, TEST, VALUE, WARNING

Entity Type Relationships/Constraints: See Section 50.2.10

- A TEXT is an entity type.
- A TEXT is a subtype of COMMON-ELEMENT.
- A TEXT may CONTAIN one or more ATTVALUE.
- A TEXT may CONTAIN one or more PCDATA.
- A TEXT may be DISPLAYED-BY any number of GRAPHIC's.
- A TEXT may be DISPLAYED-BY any number of ANNOT's.
- A TEXT may be DISPLAYED-BY any number of CAUTION's.
- A TEXT may be DISPLAYED-BY any number of CHOICE's.
- A TEXT may be DISPLAYED-BY any number of DESCINFO's.
- A TEXT may be DISPLAYED-BY any number of ENTRY's.
- A TEXT may be DISPLAYED-BY any number of FAULT's.
- A TEXT may be DISPLAYED-BY any number of FILLIN's.
- A TEXT may be DISPLAYED-BY any number of FLTSTATE's.
- A TEXT may be DISPLAYED-BY any number of GRAPHIC's.
- A TEXT may be DISPLAYED-BY any number of GRPHPRIM's.
- A TEXT may be DISPLAYED-BY any number of MENU's.
- A TEXT may be DISPLAYED-BY any number of NOTE's.
- A TEXT may be DISPLAYED-BY any number of OUTCOME's.
- A TEXT may be DISPLAYED-BY any number of PROMPT's.
- A TEXT may be DISPLAYED-BY any number of PROPERTY's.
- A TEXT may be DISPLAYED-BY any number of RECT's.
- A TEXT may be DISPLAYED-BY any number of STEP's.
- A TEXT may be DISPLAYED-BY any number of TEST's.
- A TEXT may be DISPLAYED-BY any number of VALUE's.
- A TEXT may be DISPLAYED-BY any number of WARNING's.

All TEXT's must CONTAIN a PCDATA or must CONTAIN an ATTVALUE.

128 TIME - A Label Type of ESTTIME

Data Format: 10 N R

Label Type Descriptions/Constraints: See Section 50.2.4

- A TIME is a label type with representation Numeric 10.
- A TIME may be OF at most one ESTTIME.

129 TRACK - Track Designation

Description: Represents the skill level appropriate for the
information the technician wants/requires in traversing the data,
(either Expert or Novice).

Attribute of: CONTEXT

Entity Type Relationships/Constraints: See Section 50.2.19

A TRACK is an entity type.
A TRACK may be OF any number of CONTEXT's.
A TRACK must be WITH exactly one TRACK-DATA.

130 TRACK-DATA - A Label Type of TRACK

Data Format: 10 X L

Label Type Descriptions/Constraints: See Section 50.2.19
A TRACK-DATA is a label type with representation Alphanumeric 10.
A TRACK-DATA may be OF at most one TRACK.

131 TRANSF-DATA - A Label Type of TRANSFRM

Data Format: 9 N R

Label Type Descriptions/Constraints: See Section 50.2.13
A TRANSF-DATA is a label type with representation Numeric 9.
A TRANSF-DATA may be OF at most one TRANSFRM.

132 TRANSFRM - Transformation Matrix

Description: Represents a transformation matrix which specifies coordinate translations, scalings, reflection, and rotation in terms of homogeneous coordinates.

Attribute of: GRAPHIC, GRPHPRIM

Entity Type Relationships/Constraints: See Section 50.2.13
A TRANSFRM is an entity type.
A TRANSFRM may be SPECIFIED-BY any number of GRAPHIC's.
A TRANSFRM may be SPECIFIED-BY any number of GRPHPRIM's.
A TRANSFRM must be WITH exactly one TRANSF-DATA.

133 TYPE - Element Type

Description: Represents a unique quality of an element.

Attribute of: COMMON-ELEMENT, COLHDDEF, PARTBASE

Entity Type Relationships/Constraints: See Section 50.2.12
A TYPE is an entity type.
A TYPE may be CONTAINED-IN any number of COLHDDEF's.
A TYPE may be CONTAINED-IN any number of COMMON-ELEMENT's.
A TYPE may be CONTAINED-IN any number of PARTBASE's.
A TYPE must be WITH exactly one TYPE-DATA.

134 TYPE-DATA - A Label Type of TYPE

Data Format: 80 X L

Label Type Descriptions/Constraints: See Section 50.2.12

A TYPE-DATA is a label type with representation Alphanumeric 80.
A TYPE-DATA may be OF at most one TYPE.

Example(s):

dictitem/type: "gloss", "abbsym", "symbol", or "other".
graphic/type: "normal", "locat", "overlay", "schem", "functblk",
 "wiring", "engin", or "icons"
grphprim/type: "normal", "locat", "overlay", "schem", "functblk",
 "wiring", "engin", or "icons"
system/type: "vehicle", "system" or "subsystem"
task/type: "Electrical Check of Clearing Solenoid"

135 UNITSPER - Units per Assembly, System, etc.

Description: Represents the number of units required per assembly
of a system or component.

Attribute of: PARTINFO

Entity Type Relationships/Constraints: See Section 50.2.17

An UNITSPER is an entity type.

An UNITSPER may be REQUIRED-BY any number of PARTINFO's.

An UNITSPER must be WITH exactly one UNITSPER-NUMBER.

136 UNITSPER-NUMBER - A Label Type of UNITSPER

Data Format: 10 X L

Label Type Descriptions/Constraints: See Section 50.2.17

An UNITSPER-NUMBER is a label type with representation Alphanumeric 10.

An UNITSPER-NUMBER may be OF at most one UNITSPER.

137 UOM - Unit of Measure

Description: Identifies the type of unit measurement for the
consumables needed for the current application.

Attribute of: CONSUM

Entity Type Relationships/Constraints: See Section 50.2.11

An UOM is an entity type.

An UOM may be MEASURES any number of CONSUM's.

An UOM must be WITH exactly one UOM-UNIT.

138 UOM-UNIT - A Label Type of UOM

Data Format: 15 X L

Label Type Descriptions/Constraints: See Section 50.2.11

An UOM-UNIT is a label type with representation Alphanumeric 15.

An UOM-UNIT may be OF at most one UOM.

139 USABLON - Usable on Code

Description: Represents the different configurations in which a part or assembly may appear within a system or vehicle.

Attribute of: PARTINFO

Entity Type Relationships/Constraints: See Section 50.2.17

An USABLON is an entity type.

An USABLON may IDENTIFY any number of PARTINFO's.

An USABLON must be WITH exactly one USABLON-CODE.

140 USABLON-CODE - A Label Type of USABLON

Data Format: 10 X L

Label Type Descriptions/Constraints: See Section 50.2.17

An USABLON-CODE is a label type with representation Alphanumeric 10.

An USABLON-CODE may be OF at most one USABLON.

141 USER - User Identifier

Description: Represents a user identification attribute in case the user adds annotations to the data.

Attribute of: ANNOT

Entity Type Relationships/Constraints: See Section 50.2.5

An USER is an entity type.

An USER must be WITH exactly one USER-ID.

An USER may WRITE any number of ANNOT's.

142 USER-ID - A Label Type of USER

Data Format: 10 X L

Label Type Descriptions/Constraints: See Section 50.2.5

An USER-ID is a label type with representation Alphanumeric 10.

An USER-ID may be OF at most one USER.

143 VALSTAT - Validation Status

Description: Represents whether the information has been validated or not.

Attribute of: CONTEXT

Entity Type Relationships/Constraints: See Section 50.2.19

A VALSTAT is an entity type.

A VALSTAT may be OF any number of CONTEXT's.

A VALSTAT must be WITH exactly one VALSTAT-DATA.

144 VALSTAT-DATA - A Label Type of VALSTAT

Data Format: 2 A L

Label Type Descriptions/Constraints: See Section 50.2.19

A VALSTAT-DATA is a label type with representation Character 2.

A VALSTAT-DATA may be OF at most one VALSTAT.

145 VALUE - Value

Description: References some value that may be used in a prompt or context situation.

Attribute of: ASSERTION, CHOICE, PRECOND

Entity Type Relationships/Constraints: See Section 50.2.8

A VALUE is an entity type.

A VALUE is a subtype of CDM-ELEMENT.

A VALUE must DISPLAY exactly one TEXT.

A VALUE may be ASSERTED-BY any number of ASSERTION's.

A VALUE may be CONTAINED-IN any number of CHOICE's.

A VALUE may be TESTED-BY any number of PRECOND's.

146 VERB - Verb

Description: References a common verb or action which may be performed within a task or step (e.g. remove, replace, inspect, adjust, align, etc.).

Attribute of: STEP, TASK

Entity Type Relationships/Constraints: See Section 50.2.4

A VERB is an entity type.

A VERB is a subtype of COMMON-ELEMENT.

A VERB may be USED-BY any number of STEP's.

A VERB may be USED-BY any number of TASK's.

147 VERSION - Version Number

Description: Represents what document version of the information to traverse while performing maintenance operations.

Attribute of: CONTEXT

Entity Type Relationships/Constraints: See Section 50.2.19

A VERSION is an entity type.

A VERSION may be OF any number of CONTEXT's.

A VERSION must be WITH exactly one VERSION-DATA.

148 VERSION-DATA - A Label Type of VERSION

Data Format: 10 X L

Label Type Descriptions/Constraints: See Section 50.2.19

A VERSION-DATA is a label type with representation Alphanumeric 10.

A VERSION-DATA may be OF at most one VERSION.

149 VERSTAT - Verification Status

Description: Represents whether the information has been verified or not.

Attribute of: CONTEXT

Entity Type Relationships/Constraints: See Section 50.2.19

A VERSTAT is an entity type.

A VERSTAT may be OF any number of CONTEXT's.

A VERSTAT must be WITH exactly one VERSTAT-DATA.

150 VERSTAT-DATA - A Label Type of VERSTAT

Data Format: 2 A L

Label Type Descriptions/Constraints: See Section 50.2.19

A VERSTAT-DATA is a label type with representation Character 2.

A VERSTAT-DATA may be OF at most one VERSTAT.

151 VIDEO - Video Process

Description: References an external video sequence, whether it be a file or some other external component.

Attribute of: DESCINFO, STEP

Entity Type Relationships/Constraints: See Section 50.2.9

A VIDEO is an entity type.

A VIDEO is a subtype of COMMON-ELEMENT.

A VIDEO may be INVOKED-BY any number of DESCINFO's.

A VIDEO may be INVOKED-BY any number of STEP's.

A VIDEO may REFER-TO at most one EXREFID.

A VIDEO may REFER-TO at most one FILE.

152 WARNING - Warning

Description: References a warning that is associated with the task/step which pointed to it. A warning notifies the technician that the task or step may be harmful to himself or another human if not properly performed.

Attribute of: STEP, TASK

Entity Type Relationships/Constraints: See Section 50.2.4

A WARNING is an entity type.

A WARNING is a subtype of COMMON-ELEMENT.

A WARNING must DISPLAY exactly one TEXT.
A WARNING may be REFERENCED-BY any number of STEP's.
A WARNING may be REFERENCED-BY any number of TASK's.

153 WEIGHT - Weighting Factor

Description: Represents a probability associated with a given fault within a list of faults in a fault state.

Attribute of: FLTSTATE

Entity Type Relationships/Constraints: See Section 50.2.16

A WEIGHT is an entity type.

A WEIGHT may be SCALES any number of FLTSTATE's.

A WEIGHT must be WITH exactly one WEIGHT-FACTOR.

154 WEIGHT-FACTOR - A Label Type of WEIGHT

Data Format: 10 N R

Label Type Descriptions/Constraints: See Section 50.2.16

A WEIGHT-FACTOR is a label type with representation Numeric 10.

A WEIGHT-FACTOR may be OF at most one WEIGHT.

155 WINDOW - Graphic Window

Description: Represents the size of a window to be used in clipping an illustration.

Attribute of: GRAPHIC, GRPHPRIM

Entity Type Relationships/Constraints: See Section 50.2.13

A WINDOW is an entity type.

A WINDOW may be SPECIFIED-BY any number of GRAPHIC's.

A WINDOW may be SPECIFIED-BY any number of GRPHPRIM's.

A WINDOW must be WITH exactly one WINDOW-DATA.

156 WINDOW-DATA - A Label Type of WINDOW

Data Format: 20 N R

Label Type Descriptions/Constraints: See Section 50.2.13

A WINDOW-DATA is a label type with representation Numeric 20.

A WINDOW-DATA may be OF at most one WINDOW.

157 XREF - Cross Reference

Description: References a relational link between any two elements or an element with a file or database object.

Attribute of: COMMON-ELEMENT, PARTBASE

Entity Type Relationships/Constraints: See Section 50.2.20

- A XREF is an entity type.
- A XREF is a subtype of CDM-ELEMENT.
- A XREF may CONTAIN at most one RELATION.
- A XREF may have at most one ATTNAME.
- A XREF may be REFERENCED-BY any number of COMMON-ELEMENT's.
- A XREF may be REFERENCED-BY any number of PARTBASE's.
- A XREF may REFER-TO any number of ELMNTREF's.
- A XREF may REFER-TO at most one EXREFID.

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix G - CDM Neutral Data Model Topographical Report

70.1 CDM Neutral Data Model Topographical Report

DATE: 17/ 8/90 TIME:15. 4.36

ATTVALUE

COLHDEF

```

<----->
#-----#-----#-----#0000000000#
:ATT-VALUE:ATNAME :ID_IS-REF:
:         :         :ERENCED :
#-----#-----#-----#
:
196

```

```

<-----> <----->
#-----#-----#-----#0000000000#
:ID      :COLHDEF-:COLNUM-WU:TYPE-DATA:
:         :NAME      :MBER_IS-C:_IS-CONTA:
:         :         :ONTAINED :INED      :
#-----#-----#-----#0000000000#
:
169

```

CONTEXT

```

<----->
#-----#0000000000#0000000000#0000000000#0000000000#
:ID      :MAINT-LEV:SECURITY-:VALSTAT-D:VERSTAT-D:
:         :EL       :CODE     :ATA       :ATA       :
#-----#0000000000#0000000000#0000000000#0000000000#
: : :
14 16 17
19 20 21
81

```

CONTEXT-VERSION

```

21
<----->
#-----#-----#-----#
:ID_CONTAI:VERSION-D:
:NS       :ATA      :
#-----#-----#-----#

```

CONTEXT-TRACK

```

20
<----->
#-----#-----#-----#
:ID_CONTAI:TRACK-DAT:
:NS       :A       :
#-----#-----#-----#

```

CONTEXT-RELEASE

```

19
<----->
#-----#-----#-----#
:ID_CONTAI:RELEASE-C:
:NS       :ODE      :
#-----#-----#-----#

```

CONTEXT-CONFIG

```

16
<----->
#-----#-----#-----#
:CONFIG-DA:ID_CONTAI:
:TA       :NS       :
#-----#-----#-----#

```

PARTBASE

```

<-----> <----->
#-----#0000000000#0000000000#-----#-----#0000000000#
:ID      :NSN      :ITEMID  :NAME     :PARTNUM  :REFID    :CAGE-CODE:HCI-CODE :SMR-CODE :TYPE-DATA:
:         :         :         :         :         :         :         :         :         :_IS-CONTA:
:         :         :         :         :         :         :         :         :         :INED      :
#-----#0000000000#0000000000#-----#-----#0000000000#
: :

```

XREF

```

<----->
*-----*000000000*000000000*000000000*
:ID      :ID_IS-REF:EXREF-ID:RELATION-:
:        :ERENCED  :IS-REFERE:DATA_IS-C:
:        :        :NCED      :ONTAINED :
*-----*000000000*000000000*000000000*
:
:
13 100
201

```

XREF-ATTNAME

```

201
<----->
*-----*
:ID_HAS  :ATNAME :
:        :        :
*-----*

```

PARTBASE-XREF

```

99      100
<----->
*-----*
:ID_REFERS:ID_IS-REF:
:        :ERENCED  :
*-----*

```

COMMON-ELEMENT

```

14
<----->
*-----*000000000*000000000*-----*000000000*
:ID      :ITEMID  :NAME    :REFID   :ID_IS-CON:TYPE-DATA:
:        :        :        :        :TAINED   :_IS-CONTA:
:        :        :        :        :INED     :INED      :
*-----*000000000*000000000*-----*000000000*
: : : :
1 6 7 12
15 22 43
45 48 52
53 56 59
67 76 83
89 93 101
106 110
111 118
127 167
168 171 +

```

TEXT

```

195      196
<----->
*-----*0000^0000*0000000000*
:ID      :ATT-VALUE:PCTEXT_IS:
:        :_IS-CONTA:-CONTAIN:
:        :INED      :D      :
*-----*0000000000*0000000000*
: : :
2 8 9 23
44 49 58
61 68 77
85 90 94
112 117
119 128
190 197
200

```

DICTITEM-ELEMENTREF

```

43
<----->
*-----*
:ID_REFERS:ID_IS-REF:
:          :ENCED   :
*-----*

```

COMMON-ELEM-XREF

```

12      13
<----->
*-----*
:ID_REFERS:ID_IS-REF:
:          :ERENCED  :
*-----*

```

AUDIO

```

6
<----->
*-----*0000000000*0000000000*
:ID      :EXREF-ID:FILE-DATA:
:        :IS-REFERE:_IS-REFER:
:        :NCED     :ENCED    :
*-----*0000000000*0000000000*
: :
28 133

```

CAUTION

```

7      8
<----->
*-----*
:ID      :ID_IS-DIS:
:        :PLAYED   :
*-----*
: :
135 172

```

CHOICE

```

9
<----->
*-----*
:ID      :ID_IS-DIS:
:        :PLAYED   :
*-----*
: :
10 86

```

PROPERTY

```

117
<----->
*-----*0000000000*
:ID      :ID_IS-REF:ID_IS-DIS:
:        :ERENCED  :PLAYED   :
*-----*0000000000*
: :
3 57 84
107

```

CONSUM

15

<---:--->

```

*-----*
:ID      :GOVSTD-CO:MFG-CODE:MILSPEC-C:NUMBER :UOM-UNIT:
:        :DE_SPECIF:SPECIFIES:ODE_SPECI:      :MEASURES:
:        :IES      :      :FIES      :      :
*-----*

```

: : :

137 174

DESCINFO

22

23

<---:--->

```

*-----*0000^0000*
:ID      :ID_IS-DIS:
:        :PLAYED   :
:        :
*-----*0000000000*

```

: : :

25 27 29

30 31 32

34 36 38

41 91 162

SYSTEM-DESCINFO

162

<---:---:--->

```

*-----*
:ID_IS-REF:ID_REFERS:
:ERENCED  :
*-----*

```

OPERINFO-DESCINFO

91

<---:---:--->

```

*-----*
:ID_IS-REF:ID_REFERS:
:ERENCED  :
*-----*

```

DESCINFO-DESCINFO

30

31

<---:---:--->

```

*-----*
:ID_IS-REF:ID_REFERS:
:ERENCED  :
*-----*

```

DESCINFO-AUDIO

28

29

<---:---:--->

```

*-----*
:ID_IS-INV:ID_INVOKE:
:OKED     :S
*-----*

```

ENTRY

44

<----->

```

*-----*0000000000*-----*0000^0000*
:ID      :COL-NUMBE:ID_IS-REF:ROW-NUMBE:ID_IS-DIS:
:        :R_IS-CONT:ERENCED  :R_IS-CONT:PLAYED  :
:        :AINED    :      :AINED    :
*-----*0000000000*-----*0000000000*

```

:

170

EQUIP

45

<---:--->

```

*-----*
:ID      :NUMBER
:
*-----*

```

: : :

46 47 139

176

FAULT

48

49

<---:--->

```

*-----*0000000000*0000^0000*
:ID      :MTBF-TIME:ID_IS-DIS:
:        :PLAYED   :
:        :
*-----*0000000000*0000000000*

```

: : :

50 54 62

64 120

EQUIP-ALTEQIDS

46

47

<---:---:--->

```

*-----*
:ID_ALT-EQ:ID_HAS-AL:
:UIP      :T
*-----*

```

FAULTINF-FAULT

```

54      52
<----->
*-----*
:ID_IS-IDE:ID_IDENTI:
:NTIFIED :FIES :
: : :
*-----*
:
163

```

TEST

```

189      190
<----->
*-----*000000000*000000000*0000*0000*
:ID :AGENT-ID:RANGE-DAT:ID_IS-DIS:
: :PERFORMS :A_REPRESE:PLAYED :
: : :NTS :
*-----*000000000*000000000*000000000*
: : :
55 60 125
192 194

```

FILLIN

```

56      57      58
<----->
*-----*000000000*-----*000000000*-----*
:ID :DEFAULT-D:ID_IS-ASS:RANGE-DAT:ID_IS-DIS:
: :ATA :ERTED :A :PLAYED :
*-----*000000000*-----*000000000*-----*
:
113

```

FLTSTATE

```

59      60      61
<----->
*-----*0000*0000*0000*0000*
:ID :ID_IS-IDE:ID_IS-DIS:
: :NTIFIED :PLAYED :
*-----*000000000*000000000*
: : :
63 65 66
95

```

FLTSTATE-WEIGHT

```

66
<----->
*-----*
:ID_SCALED:WEIGHT-FA:
: :CTOR_SCAL:
: :ES :
*-----*

```

FLTSTATE-IMPFAULT

```

64      65
<----->
*-----*
:ID_IMPLIC:ID_IMPLIC:
:ATING :ATED :
: : :
*-----*

```

FLTSTATE-EXPFAULT

```

62      63
<----->
*-----*
:ID_EXCULP:ID_EXCULP:
:ATING :ATED :
: : :
*-----*

```

GRAPHIC

```

67      68
<----->
*-----*000000000*000000000*0000*0000*
:ID :PENPATT-D:PENSHAPE-:ID_IS-DIS:
: :ATA_IS-SP:DATA_IS-S:PALYED :
: :ECIFIES :PECIFIED :
*-----*000000000*000000000*000000000*
: : :
33 69 70
71 73 74
75 102
141

```

GRAPHIC-WINDOW

75

```

<----->
*-----*
:ID_SPECIF:WINDOW-DA:
:IES      :TA_IS-SPE:
:         :CIFIED   :
*-----*

```

GRAPHIC-TRANSFORM

74

```

<----->
*-----*
:ID_SPECIF:TRANSF-DA:
:IES      :TA_IS-SPE:
:         :CIFIED   :
*-----*

```

GRAPHIC-MINSIZE

73

```

<----->
*-----*
:ID_SPECIF:MINSIZE-C:
:IES      :ODE_IS-SP:
:         :ECIFIES  :
*-----*

```

GRAPHIC-FOCUS

69

70

```

<----->
*-----*
:ID_IS-FOC:ID_SPECIF:
:USED     :IES-FOCUS:
:         :         :
*-----*

```

DESCINFO-GRAPHIC

32

33

```

<----->
*-----*
:ID_DISPLA:ID_IS-DIS:
:YS        :PLAYED  :
:         :         :
*-----*

```

GRPHPRIM

76

77

```

<----->
*-----*000000000*000000000*000000000*000000000*000000000*000000000*000000000*000000000*000000000*
:ID        :CODING-CO:FILE-DATA:PCTEXT_IS:PEWPATT-D:PEWSHAPE-:START-DAT:STOP-DATA:ID_IS-DIS:
:         :DE_IS-SPE:_IS-REFER:-CONTAIN:ATA_IS-SP:DATA_IS-S:A         :PLAYED  :
:         :CIFIED   :ENCED   :D         :ECIFIED  :PECIFIED  :         :         :
*-----*000000000*000000000*000000000*000000000*000000000*000000000*000000000*000000000*000000000*
: : :
72 78 79
80

```

GRPHPRIM-WINDOW

80

```

<----->
*-----*
:ID_SPECIF:WINDOW-DA:
:IES      :TA_IS-SPE:
:         :CIFIED   :
*-----*

```

GRPHPRIM-TRANSFORM

79

```

<----->
*-----*
:ID_SPECIF:TRANSF-DA:
:IES      :TA_IS-SPE:
:         :CIFIED   :
*-----*

```

GRPHPRIM-MINSIZE

78

```

<----->
*-----*
:ID_SPECIF:MINSIZE-C:
:IES      :ODE_IS-SP:
:         :ECIFIED  :
*-----*

```

GRAPHIC-GRPGPRIM

71

72

```

<----->
*-----*
:ID_INVOKE:ID_IS-INV:
:S         :OKED    :
:         :         :
*-----*

```

LIST

81

```

<----->
*-----*000000000*-----*
:ID        :LIST-NAME:REFID   :ID_IS-COM:
:         :         :         :TAINED   :
*-----*000000000*-----*
: : :
82

```

MENU

83

84

85

```

<----->
*-----*000000000*-----*
:ID        :ID_IS-ASS:SELECT-DA:ID_IS-DIS:
:         :ERTED   :TA        :PLAYED  :
*-----*000000000*-----*
: : :
87 88 115

```

NOTE

89

90

```

<----->
*-----*
:ID        :ID_IS-DIS:
:         :PLAYED  :
*-----*
: : :
143 179

```

MENU-DEFAULT

```

      88
<----->
*-----*
:DEFAULT-D:ID_HAS :
:ATA      :
*-----*

```

MENU-CHOICE

```

      86      87
<----->
*-----*
:ID      ID_HAS :
:        :
*-----*

```

LIST-ELMNTREF

```

      82
<----->
*-----*
:ID_IS-REF:ID_REFERS:
:ERENCED :
*-----*

```

OUTCOME

```

      93      94
<----->
*-----*000000000*
:ID      ID_IS-DIS:
:        PLAYED :
*-----*000000000*
: : :
96 97 191

```

TEST-OUTCOME

```

      191      192
<----->
*-----*
:ID_IS-PRO:ID_PRODUC:
:DUCE     ES :
*-----*

```

OUTCOME-FLTSTATE

```

      95      96
<----->
*-----*
:ID_IS-IDE:ID_IDENTI:
:NTIFIED  FIES :
*-----*

```

PARTINFO

```

      101
<-----> <----->
*-----*000000000*000000000*-----*000000000*-----*
:ID      REFDES :WOUNID :WOUNTYPE :INDX-NUMB:MTBF-TIME:REP-LEVEL:UNITSPER:USABLOW-C:
:        :        :        :ER_REFERS:      _IS-IDENT:NUMBER_IS:ODE_IDENT:
:        :        :        :        :        IFIED  -REQUIRED:IFIES :
*-----*000000000*000000000*-----*000000000*-----*
: :
81 103
105 165

```

SYSTEM-PARTINFO

```

      165
<----->
*-----*
:ID_IS-IDE:ID_IDENTI:
:NTIFIED  FIES :
*-----*

```

PARTINFO-PARTBASE

```

      104      105
<----->
*-----*
:ID_IS-IDE:ID_IDENTI:
:NTIFIED  FIES :
*-----*

```

PARTINFO-GRAPHIC

```

      102      103
<----->
*-----*
:ID_IS-IDE:ID_IDENTI:
:NTIFIED  FIES :
*-----*

```

FAULT-PARTINFO

```

      50      51
<----->
*-----*
:ID_POINTS:ID_IS-POI:
:          NTED-TO :
*-----*

```


PERSON

106

<----->

```

*-----*
:ID      :NUMBER:
:        :      :
:        :      :
:        :      :
*-----*

```

: :
145 181

PRECOND

107

<----->

```

*-----*000000000*000000000*-----*
:ID      :OP-DATA :POLARITY-:ID_IS-TES:
:        :        :DATA      :TED      :
:        :        :        :        :
:        :        :        :        :
*-----*000000000*000000000*-----*

```

: : :
18 98 108
126

PROCESS

110

<----->

```

*-----*000000000*000000000*-----*
:ID      :EXREF-ID:FILE-DATA:
:        :IS-REFER:IS-REFER:
:        :NCED     :EMCED     :
:        :        :        :
*-----*000000000*000000000*-----*

```

: :
35 147

REQCOND-PRECOND

126

<----->

```

*-----*
:ID_IS-TES:ID_TESTS:
:TED      :        :
*-----*

```

OUTCOME-PRECOND

97

98

<----->

```

*-----*
:ID_TESTS :ID_IS-TES:
:        :TED      :
*-----*

```

DESCINFO-PROCESS

34

35

<----->

```

*-----*
:ID_INVOKE:ID_IS-INV:
:IS        :OKED     :
*-----*

```

CONTEXT-PRECOND

17

18

<----->

```

*-----*
:ID_TESTS :ID_IS-TES:
:        :TED      :
*-----*

```

PROMPT

111

112

<----->

```

*-----*0000*0000*
:ID      :ID_IS-DIS:
:        :PLAYED   :
:        :        :
*-----*000000000*

```

: :
37 114
116 149

ASSERTION

3

<----->

```

*-----*
:ID      :ID_IS-ASS:
:        :ERTED    :
:        :        :
*-----*

```

: : :
4 26 131

RECT

118

119

<----->

```

*-----*000000000*000000000*0000*0000*
:ID      :ACTION-CO:AGENT-ID :ID_IS-DIS:
:        :DE_IS-IDE:PERFORMS :PLAYED   :
:        :NTIFIED   :        :
*-----*000000000*000000000*000000000*

```

: :
121 122
124

RECT-TEST

124

125

<----->

```

*-----*
:ID_IDENTI:ID_IS-IDE:
:FIES      :NTIFIED :
*-----*

```

RECT-FAULT

120

121

<----->

```

*-----*
:ID_IDENTI:ID_IS-IDE:
:FIES      :NTIFIED :
*-----*

```

PROMPT-MENU

115

116

<----->

```

*-----*
:ID_CALLED:ID_CALLS :
:        :        :
*-----*

```

PROMPT-FILLIN

113

114

<----->

```

*-----*
:ID      :ID_CONTAI:
:        :NS        :
*-----*

```

DESCINFO-PROMPT

36

37

<----->

```

*-----*
:ID_PRESEN:ID_IS-PRE:
:TS        :SENTED   :
*-----*

```

DESCINFO-ASSERTION

26 27

```

<----->
*-----*
:ID_IS-ASS:ID_ASSERT:
:ERTED :S :
*-----*

```

STEP

127

128

TABLE-COLHDEF

169

167

TABLE-ENTRY

170

168

TASK

171

```

<-----> : <-----> <-----> <----->
*-----* *-----* *-----* *-----*
:ID :TIME_IS-R:ID_IS-DIS: :ID_IS-COM:ID_CONTAI: :ID_IS-COM:ID_CONTAI: :ID :TIME_IS-R:
: :REQUIRED :PLAYED : :TAINED :NS : :TAINED :NS : :REQUIRED :
*-----* *-----* *-----* *-----*
: : : : : :
130 132 39 155 40 156 92 123
134 136
138 140
142 144
146 148
150 151
152 153
154 157
158 160
184

```

TEST-TASK

193

194

TASK-VERB

186

TASK-STEP

184

185

TASK-REQCOND

183

TASK-PERSON

181

182

```

<-----> <-----> <-----> <-----> <----->
*-----* *-----* *-----* *-----* *-----*
:ID_IS-IDE:ID_IDENTI: :ID_USES :ID_IS-USE: :ID_IS-REF:ID_REFERS: :ID_IS-TES:ID_TESTS : :ID_IS-REQ:ID_REQUIR:
:NTIFIED :FIES : : :TAINED :NS : :TAINED :NS : :UIRED :ES :
*-----* *-----* *-----* *-----* *-----*

```

TASK-NOTE

179

180

TASK-FOLLOWON

178

TASK-EQUIP

176

177

TASK-CONSUM

174

175

TASK-CAUTION

172

173

```

<-----> <-----> <-----> <-----> <----->
*-----* *-----* *-----* *-----* *-----*
:ID_IS-REF:ID_REFERS: :REFID_FOL:ID_FOLLOW: :ID_IS-REQ:ID_REQUIR: :ID_IS-REQ:ID_REQUIR: :ID_IS-REF:ID_REFERS:
:ERENCED : : :LOWING :ED : :UIRED :ES : :UIRED :ES : :ERENCED :
*-----* *-----* *-----* *-----* *-----*

```

SYSTEM-TASK 166	STEP-VERB 157	STEP-TABLE 154 155 156	STEP-SUBSTEP 152 153	STEP-REQCOND 151
<----->	<----->	<----->	<----->	<----->
-----	*-----*	*-----*	*-----*	*-----*
:ID_REFERS:ID_IS-REF:	:ID_USES :ID_IS-USE:	:ID_PRESEN:ID_IS-PRE:	:ID_IS-REF:ID_REFERS:	:ID_IS-TEST:ID_TESTS :
: :ERENCED :	: :D :	:TS :SENTED :	:ERENCED :	:TED :
-----	*-----*	*-----*	*-----*	*-----*

STEP-PROMPT 149 150	STEP-PROCESS 147 148	STEP-PERSON 145 146	STEP-NOTE 143 144	STEP-GRAPHIC 141 142
<----->	<----->	<----->	<----->	<----->
-----	*-----*	*-----*	*-----*	*-----*
:ID_IS-PRE:ID_PRESEN:	:ID_IS-INV:ID_INVOKE:	:ID_IS-REQ:ID_REQUIR:	:ID_IS-REF:ID_REFERS:	:ID_IS-DIS:ID_DISPLA:
:SENTED :TS :	:OKED :S :	:UIRED :ES :	:ERENCED :	:PLAYED :YS :
-----	*-----*	*-----*	*-----*	*-----*

STEP-EQUIP 139 140	STEP-COMSUM 137 138	STEP-CAUTION 135 136	STEP-AUDIO 133 134	STEP-ASSERTION 131 132
<----->	<----->	<----->	<----->	<----->
-----	*-----*	*-----*	*-----*	*-----*
:ID_IS-REQ:ID_REQUIR:	:ID_IS-REQ:ID_REQUIR:	:ID_IS-REF:ID_REFERS:	:ID_IS-INV:ID_INVOKE:	:ID_IS-ASS:ID_ASSERT:
:UIRED :ES :	:UIRED :ES :	:ERENCED :	:OKED :S :	:ERTED :S :
-----	*-----*	*-----*	*-----*	*-----*

RECT-TASK
122 123

<----->

:ID_IDENTI:ID_IS-IDE:

:FIES :NTIFIED :

OPERINFO-TASK
92

<----->

:ID_REFERS:ID_IS-REF

: :ERENCED :

DESCINFO-TABLE
38 39 40

<----->

:ID_PRESEN:ID_IS-PRE:

:TS :SENTED :

FAULTINF-TEST
53 55

<----->

:ID_IDENTI:ID_IS-IDE:

:FIES :NTIFIED :

: :

164

ANNOT
1 2

<----->

:ID :ID_IS-DIS:USER-ID_W:

: :PLAYED :RITES :

: :

24 129

VALUE
197

<----->

:ID :ID_IS-DIS:

: :PLAYED :

: : :

5 11 109

VIDEO
198

<----->

:ID :EXREF-ID :FILE-DATA:

: :IS-REFERE:IS-REFER:

: :NCED :ENCED :

: :

42 159

SYSTEM-FAULTINF

163 164

<----->

```

*-----*
:ID_IS-REF:ID_REFERS:
:ERENCED :
*-----*

```

STEP-VIDEO

158 159

<----->

```

*-----*
:ID_INVOKE:ID_IS-INV:
:S :OKED :
*-----*

```

STEP-ANNOT

129 130

<----->

```

*-----*
:ID_IS-COM:ID_CONTAI:
:TAINED :NS :
*-----*

```

PRECOND-VALUE

108 109

<----->

```

*-----*
:ID_TESTS :ID_IS-TES:
: :TED :
*-----*

```

DESCINFO-VIDEO

41 42

<----->

```

*-----*
:ID_INVOKE:ID_IS-INV:
:S :OKED :
*-----*

```

DESCINFO-ANNOT

24 25

<----->

```

*-----*
:ID_IS-REF:ID_REFERS:
:ERENCED :
*-----*

```

WARNING

199 200

<----->

```

*-----*
:ID :ID_IS-DIS:
: :PLAYED :
*-----*

```

:

161 188

ASSERTION--VALUE

4 5

<----->

```

*-----*
:ID_ASSERT:ID_IS-ASS:
:S :ERTED :
*-----*

```

CHOICE-VALUE

10 11

<----->

```

*-----*
:ID_CONTAI:ID_IS-COM:
:WS :TAINED :
*-----*

```

TASK-WARNING

187 188

<----->

```

*-----*
:ID_REFERS:ID_IS-REF:
: :ERENCED :
*-----*

```

STEP-WARNING

160 161

<----->

```

*-----*
:ID_REFERS:ID_IS-REF:
: :ERENCED :
*-----*

```

REQCOND-ELMNTREF

<----->

```

*-----*
:ID_IS-REF:ID_REFERS:
:ERENCED :
*-----*

```

SYSTEM-OPERINFO

<----->

```

*-----*
:ID_IS-REF:ID_REFERS:
:ERENCED :
*-----*

```

SYSTEM-SUBSYSTEM

<----->

```

*-----*
:ID_IS-REF:ID_REFERS:
:ERENCED :
*-----*

```

TECHINFO-SYSTEM

<----->

```

*-----*
:ID_IS-COM:ID_CONTAI:
:TAINED :NS :
*-----*

```

70.2 Cross Reference of the CDM Neutral Data Model

PATH	FROM STRUCTURE	TO STRUCTURE	SUBSET CONSTRAINT	EQUAL.CONSTRAINT
----	-----	-----	-----	-----
1	ANNOT	COMMON-ELEMENT	generated	
2	ANNOT	TEXT	generated	
3	ASSERTION	PROPERTY	generated	
4	ASSERTION--VALUE	ASSERTION		generated
5	ASSERTION--VALUE	VALUE	generated	
6	AUDIO	COMMON-ELEMENT	generated	
7	CAUTION	COMMON-ELEMENT	generated	
8	CAUTION	TEXT	generated	
9	CHOICE	TEXT	generated	
10	CHOICE-VALUE	CHOICE		generated
11	CHOICE-VALUE	VALUE	generated	
12	COMMON-ELEM-XREF	COMMON-ELEMENT	generated	
13	COMMON-ELEM-XREF	XREF	generated	
14	COMMON-ELEMENT	CONTEXT	generated	
15	CONSUM	COMMON-ELEMENT	generated	
16	CONTEXT-CONFIG	CONTEXT	generated	
17	CONTEXT-PRECOND	CONTEXT	generated	
18	CONTEXT-PRECOND	PRECOND	generated	
19	CONTEXT-RELEASE	CONTEXT	generated	
20	CONTEXT-TRACK	CONTEXT	generated	
21	CONTEXT-VERSION	CONTEXT	generated	
22	DESCINFO	COMMON-ELEMENT	generated	
23	DESCINFO	TEXT	generated	
24	DESCINFO-ANNOT	ANNOT	generated	
25	DESCINFO-ANNOT	DESCINFO	generated	
26	DESCINFO-ASSERTION	ASSERTION	generated	
27	DESCINFO-ASSERTION	DESCINFO	generated	
28	DESCINFO-AUDIO	AUDIO	generated	
29	DESCINFO-AUDIO	DESCINFO	generated	
30	DESCINFO-DESCINFO	DESCINFO	generated	
31	DESCINFO-DESCINFO	DESCINFO	generated	
32	DESCINFO-GRAPHIC	DESCINFO	generated	
33	DESCINFO-GRAPHIC	GRAPHIC	generated	
34	DESCINFO-PROCESS	DESCINFO	generated	
35	DESCINFO-PROCESS	PROCESS	generated	
36	DESCINFO-PROMPT	DESCINFO	generated	
37	DESCINFO-PROMPT	PROMPT	generated	
38	DESCINFO-TABLE	DESCINFO	generated	
39	DESCINFO-TABLE	TABLE-COLHDDEF	generated	
40	DESCINFO-TABLE	TABLE-ENTRY	generated	
41	DESCINFO-VIDEO	DESCINFO	generated	
42	DESCINFO-VIDEO	VIDEO	generated	
43	DICTITEM-ELMNTREF	COMMON-ELEMENT	generated	
44	ENTRY	TEXT	generated	
45	EQUIP	COMMON-ELEMENT	generated	
46	EQUIP-ALTEQIDS	EQUIP	generated	
47	EQUIP-ALTEQIDS	EQUIP	generated	

48	FAULT	COMMON-ELEMENT	generated	
49	FAULT	TEXT	generated	
50	FAULT-PARTINFO	FAULT	generated	
51	FAULT-PARTINFO	PARTINFO	generated	
52	FAULTINF-FAULT	COMMON-ELEMENT	generated	
53	FAULTINF-TEST	COMMON-ELEMENT	generated	
54	FAULTINF-FAULT	FAULT	generated	
55	FAULTINF-TEST	TEST	generated	
56	FILLIN	COMMON-ELEMENT	generated	
57	FILLIN	PROPERTY	generated	
58	FILLIN	TEXT	generated	
59	FLTSTATE	COMMON-ELEMENT	generated	
60	FLTSTATE	TEST	generated	
61	FLTSTATE	TEXT	generated	
62	FLTSTATE-EXPFAULT	FAULT	generated	
63	FLTSTATE-EXPFAULT	FLTSTATE	generated	
64	FLTSTATE-IMPFAULT	FAULT	generated	
65	FLTSTATE-IMPFAULT	FLTSTATE	generated	
66	FLTSTATE-WEIGHT	FLTSTATE	generated	
67	GRAPHIC	COMMON-ELEMENT	generated	
68	GRAPHIC	TEXT	generated	
69	GRAPHIC-FOCUS	GRAPHIC	generated	
70	GRAPHIC-FOCUS	GRAPHIC	generated	
71	GRAPHIC-GRPGPRIM	GRAPHIC		generated
72	GRAPHIC-GRPGPRIM	GRPHPRIM	generated	
73	GRAPHIC-MINSIZE	GRAPHIC	generated	
74	GRAPHIC-TRANSFORM	GRAPHIC	generated	
75	GRAPHIC-WINDOW	GRAPHIC	generated	
76	GRPHPRIM	COMMON-ELEMENT	generated	
77	GRPHPRIM	TEXT	generated	
78	GRPHPRIM-MINSIZE	GRPHPRIM	generated	
79	GRPHPRIM-TRANSFORM	GRPHPRIM	generated	
80	GRPHPRIM-WINDOW	GRPHPRIM	generated	
81	LIST	CONTEXT	generated	
82	LIST-ELMNTREF	LIST		generated
83	MENU	COMMON-ELEMENT	generated	
84	MENU	PROPERTY	generated	
85	MENU	TEXT	generated	
86	MENU-CHOICE	CHOICE	generated	
87	MENU-CHOICE	MENU		generated
88	MENU-DEFAULT	MENU	generated	
89	NOTE	COMMON-ELEMENT	generated	
90	NOTE	TEXT	generated	
91	OPERINFO-DESCINFO	DESCINFO	generated	
92	OPERINFO-TASK	TASK	generated	
93	OUTCOME	COMMON-ELEMENT	generated	
94	OUTCOME	TEXT	generated	
95	OUTCOME-FLTSTATE	FLTSTATE	generated	
96	OUTCOME-FLTSTATE	OUTCOME		generated
97	OUTCOME-PRECOND	OUTCOME		generated
98	OUTCOME-PRECOND	PRECOND	generated	
99	PARTBASE-XREF	PARTBASE	generated	

100	PARTBASE-XREF	XREF	generated	
101	PARTINFO	COMMON-ELEMENT	generated	
102	PARTINFO-GRAPHIC	GRAPHIC	generated	
103	PARTINFO-GRAPHIC	PARTINFO		generated
104	PARTINFO-PARTBASE	PARTBASE	generated	
105	PARTINFO-PARTBASE	PARTINFO		generated
106	PERSON	COMMON-ELEMENT	generated	
107	PRECOND	PROPERTY	generated	
108	PRECOND-VALUE	PRECOND		generated
109	PRECOND-VALUE	VALUE	generated	
110	PROCESS	COMMON-ELEMENT	generated	
111	PROMPT	COMMON-ELEMENT	generated	
112	PROMPT	TEXT	generated	
113	PROMPT-FILLIN	FILLIN	generated	
114	PROMPT-FILLIN	PROMPT	generated	
115	PROMPT-MENU	MENU	generated	
116	PROMPT-MENU	PROMPT	generated	
117	PROPERTY	TEXT	generated	
118	RECT	COMMON-ELEMENT	generated	
119	RECT	TEXT	generated	
120	RECT-FAULT	FAULT		generated
121	RECT-FAULT	RECT	generated	
122	RECT-TASK	RECT		generated
123	RECT-TASK	TASK	generated	
124	RECT-TEST	RECT	generated	
125	RECT-TEST	TEST	generated	
126	REQCOND-PRECOND	PRECOND	generated	
127	STEP	COMMON-ELEMENT	generated	
128	STEP	TEXT	generated	
129	STEP-ANNOT	ANNOT	generated	
130	STEP-ANNOT	STEP	generated	
131	STEP-ASSERTION	ASSERTION	generated	
132	STEP-ASSERTION	STEP	generated	
133	STEP-AUDIO	AUDIO	generated	
134	STEP-AUDIO	STEP	generated	
135	STEP-CAUTION	CAUTION	generated	
136	STEP-CAUTION	STEP	generated	
137	STEP-COMSUM	CONSUM	generated	
138	STEP-COMSUM	STEP	generated	
139	STEP-EQUIP	EQUIP	generated	
140	STEP-EQUIP	STEP	generated	
141	STEP-GRAPHIC	GRAPHIC	generated	
142	STEP-GRAPHIC	STEP	generated	
143	STEP-NOTE	NOTE	generated	
144	STEP-NOTE	STEP	generated	
145	STEP-PERSON	PERSON	generated	
146	STEP-PERSON	STEP	generated	
147	STEP-PROCESS	PROCESS	generated	
148	STEP-PROCESS	STEP	generated	
149	STEP-PROMPT	PROMPT	generated	
150	STEP-PROMPT	STEP	generated	
151	STEP-REQCOND	STEP	generated	

152	STEP-SUBSTEP	STEP	generated	
153	STEP-SUBSTEP	STEP	generated	
154	STEP-TABLE	STEP	generated	
155	STEP-TABLE	TABLE-COLHDDEF	generated	
156	STEP-TABLE	TABLE-ENTRY	generated	
157	STEP-VERB	STEP	generated	
158	STEP-VIDEO	STEP	generated	
159	STEP-VIDEO	VIDEO	generated	
160	STEP-WARNING	STEP	generated	
161	STEP-WARNING	WARNING	generated	
162	SYSTEM-DESCINFO	DESCINFO	generated	
163	SYSTEM-FAULTINF	FAULTINF-FAULT	generated	
164	SYSTEM-FAULTINF	FAULTINF-TEST	generated	
165	SYSTEM-PARTINFO	PARTINFO	generated	
166	SYSTEM-TASK	TASK	generated	
167	TABLE-COLHDDEF	COMMON-ELEMENT	generated	
168	TABLE-ENTRY	COMMON-ELEMENT	generated	
169	TABLE-COLHDDEF	COLHDDEF	generated	
170	TABLE-ENTRY	ENTRY	generated	
171	TASK	COMMON-ELEMENT	generated	
172	TASK-CAUTION	CAUTION	generated	
173	TASK-CAUTION	TASK	generated	
174	TASK-CONSUM	CONSUM	generated	
175	TASK-CONSUM	TASK	generated	
176	TASK-EQUIP	EQUIP	generated	
177	TASK-EQUIP	TASK	generated	
178	TASK-FOLLOWON	TASK	generated	
179	TASK-NOTE	NOTE	generated	
180	TASK-NOTE	TASK	generated	
181	TASK-PERSON	PERSON	generated	
182	TASK-PERSON	TASK	generated	
183	TASK-REQCOND	TASK	generated	
184	TASK-STEP	STEP	generated	
185	TASK-STEP	TASK		generated
186	TASK-VERB	TASK	generated	
187	TASK-WARNING	TASK	generated	
188	TASK-WARNING	WARNING	generated	
189	TEST	COMMON-ELEMENT	generated	
190	TEST	TEXT	generated	
191	TEST-OUTCOME	OUTCOME	generated	
192	TEST-OUTCOME	TEST		generated
193	TEST-TASK	TASK	generated	
194	TEST-TASK	TEST		generated
195	TEXT	COMMON-ELEMENT	generated	
196	TEXT	ATTVALUE	generated	
197	VALUE	TEXT	generated	
198	VIDEO	COMMON-ELEMENT	generated	
199	WARNING	COMMON-ELEMENT	generated	
200	WARNING	TEXT	generated	
201	XREF-ATTNAME	XREF	generated	

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix H - CDM/ORACLE-SQL Database Schema.

80.1 CDM/ORACLE-SQL Database Schema.

```

CREATE TABLE ANNOT          (ID                      char(10),
                             ID_IS_DISPLAYED         char(10) not null,
                             USER_ID_WRITES          char(10));

CREATE UNIQUE INDEX IDX1 ON ANNOT          (ID);

CREATE TABLE ASSERTION      (ID                      char(10),
                             ID_IS_ASSERTED          char(10) not null);

CREATE UNIQUE INDEX IDX2 ON ASSERTION      (ID);

CREATE TABLE ASSERTION__VALUE (ID_ASSERTS           char(10) not null,
                             ID_IS_ASSERTED          char(10) not null);

CREATE UNIQUE INDEX IDX3 ON ASSERTION__VALUE (ID_ASSERTS,
                                                ID_IS_ASSERTED);

CREATE TABLE ATTVALUE       (ATT_VALUE             char(10) not null,
                             ATNAME                  char(15) not null,
                             ID_IS_REFERENCED        char(10) not null);

CREATE UNIQUE INDEX IDX4 ON ATTVALUE       (ATT_VALUE);

CREATE TABLE AUDIO          (ID                      char(10),
                             EXREF_ID_IS_REFERENCED  char(10),
                             FILE_DATA_IS_REFERENCED char(80));

CREATE UNIQUE INDEX IDX5 ON AUDIO          (ID);

CREATE TABLE CAUTION        (ID                      char(10),
                             ID_IS_DISPLAYED         char(10) not null);

CREATE UNIQUE INDEX IDX6 ON CAUTION        (ID);

CREATE TABLE CHOICE          (ID                      char(10),
                             ID_IS_DISPLAYED         char(10) not null);

CREATE UNIQUE INDEX IDX7 ON CHOICE          (ID);

CREATE TABLE CHOICE__VALUE  (ID_CONTAINS           char(10) not null,
                             ID_IS_CONTAINED         char(10) not null);

CREATE UNIQUE INDEX IDX8 ON CHOICE__VALUE  (ID_CONTAINS,
                                                ID_IS_CONTAINED);

CREATE TABLE COLHDDEF       (ID                      char(10),
                             COLHDDEF_NAME           char(20) not null,
                             COLNUM_NUMBER_IS_CONTAINED number (4) not null,
                             TYPE_DATA_IS_CONTAINED  char(80));

CREATE UNIQUE INDEX IDX9 ON COLHDDEF       (ID);

CREATE UNIQUE INDEX IDX10 ON COLHDDEF      (COLHDDEF_NAME);

CREATE TABLE COMMON_ELEM_XREF (ID_REFERS           char(10) not null,
                             ID_IS_REFERENCED        char(10) not null);

CREATE UNIQUE INDEX IDX11 ON COMMON_ELEM_XREF (ID_REFERS,
                                                ID_IS_REFERENCED);

CREATE TABLE COMMON_ELEMENT (ID                      char(10),
                             ITEMID                  char(10),
                             NAME                     char(20),
                             REFID                    char(10) not null,
                             ID_IS_CONTAINED         char(10) not null,
                             TYPE_DATA_IS_CONTAINED  char(80));

CREATE UNIQUE INDEX IDX12 ON COMMON_ELEMENT (ID);

CREATE TABLE CONSUM          (ID                      char(10),
                             GOVSTD_CODE_SPECIFIES   char(20) not null,

```

	MFG_CODE_SPECIFIES	char(20) not null,
	MILSPEC_CODE_SPECIFIES	char(20) not null,
	NUMBER1	number (10) not null,
	UOM_UNIT_MEASURES	char(15) not null);
CREATE UNIQUE INDEX IDX13 ON CONSUM	(ID);	
CREATE TABLE CONTEXT	(ID	char(10),
	MAINT_LEVEL	char(10),
	SECURITY_CODE	char(10),
	VALSTAT_DATA	char(10),
	VERSTAT_DATA	char(10));
CREATE UNIQUE INDEX IDX14 ON CONTEXT	(ID);	
CREATE TABLE CONTEXT_CONFIG	(CONFIG_DATA	char(20) not null,
	ID_CONTAINS	char(10) not null);
CREATE UNIQUE INDEX IDX15 ON CONTEXT_CONFIG	(CONFIG_DATA,	
	ID_CONTAINS);	
CREATE TABLE CONTEXT_PRECOND	(ID_TESTS	char(10) not null,
	ID_IS_TESTED	char(10) not null);
CREATE UNIQUE INDEX IDX16 ON CONTEXT_PRECOND	(ID_TESTS,	
	ID_IS_TESTED);	
CREATE TABLE CONTEXT_RELEASE	(ID_CONTAINS	char(10) not null,
	RELEASE_CODE	char(10) not null);
CREATE UNIQUE INDEX IDX17 ON CONTEXT_RELEASE	(ID_CONTAINS,	
	RELEASE_CODE);	
CREATE TABLE CONTEXT_TRACK	(ID_CONTAINS	char(10) not null,
	TRACK_DATA	char(10) not null);
CREATE UNIQUE INDEX IDX18 ON CONTEXT_TRACK	(ID_CONTAINS,	
	TRACK_DATA);	
CREATE TABLE CONTEXT_VERSION	(ID_CONTAINS	char(10) not null,
	VERSION_DATA	char(10) not null);
CREATE UNIQUE INDEX IDX19 ON CONTEXT_VERSION	(ID_CONTAINS,	
	VERSION_DATA);	
CREATE TABLE DESCINFO	(ID	char(10),
	ID_IS_DISPLAYED	char(10));
CREATE UNIQUE INDEX IDX20 ON DESCINFO	(ID);	
CREATE TABLE DESCINFO_ANNOT	(ID_IS_REFERENCED	char(10) not null,
	ID_REFERS	char(10) not null);
CREATE UNIQUE INDEX IDX21 ON DESCINFO_ANNOT	(ID_IS_REFERENCED,	
	ID_REFERS);	
CREATE TABLE DESCINFO_ASSERTION	(ID_IS_ASSERTED	char(10) not null,
	ID_ASSERTS	char(10) not null);
CREATE UNIQUE INDEX IDX22 ON DESCINFO_ASSERTION	(ID_IS_ASSERTED,	
	ID_ASSERTS);	
CREATE TABLE DESCINFO_AUDIO	(ID_IS_INVOKED	char(10) not null,
	ID_INVOKES	char(10) not null);
CREATE UNIQUE INDEX IDX23 ON DESCINFO_AUDIO	(ID_IS_INVOKED,	
	ID_INVOKES);	
CREATE TABLE DESCINFO_DESCINFO	(ID_IS_REFERENCED	char(10) not null,
	ID_REFERS	char(10) not null);
CREATE UNIQUE INDEX IDX24 ON DESCINFO_DESCINFO	(ID_IS_REFERENCED,	
	ID_REFERS);	
CREATE TABLE DESCINFO_GRAPHIC	(ID_DISPLAYS	char(10) not null,
	ID_IS_DISPLAYED	char(10) not null);

```

CREATE UNIQUE INDEX IDX25 ON DESCINFO_GRAPHIC (ID_DISPLAYS,
                                                ID_IS_DISPLAYED);
CREATE TABLE DESCINFO_PROCESS (ID_INVOKES char(10) not null,
                                ID_IS_INVOKED char(10) not null);
CREATE UNIQUE INDEX IDX26 ON DESCINFO_PROCESS (ID_INVOKES,
                                                ID_IS_INVOKED);
CREATE TABLE DESCINFO_PROMPT (ID_PRESENTS char(10) not null,
                                ID_IS_PRESENTED char(10) not null);
CREATE UNIQUE INDEX IDX27 ON DESCINFO_PROMPT (ID_PRESENTS,
                                                ID_IS_PRESENTED);
CREATE TABLE DESCINFO_TABLE (ID_PRESENTS char(10) not null,
                                ID_IS_PRESENTED char(10) not null);
CREATE UNIQUE INDEX IDX28 ON DESCINFO_TABLE (ID_PRESENTS,
                                                ID_IS_PRESENTED);
CREATE TABLE DESCINFO_VIDEO (ID_INVOKES char(10) not null,
                                ID_IS_INVOKED char(10) not null);
CREATE UNIQUE INDEX IDX29 ON DESCINFO_VIDEO (ID_INVOKES,
                                                ID_IS_INVOKED);
CREATE TABLE DICTITEM_ELMNTREF (ID_REFERS char(10) not null,
                                ID_IS_REFENCED char(10) not null);
CREATE UNIQUE INDEX IDX30 ON DICTITEM_ELMNTREF (ID_REFERS,
                                                ID_IS_REFENCED);
CREATE TABLE ENTRY (ID char(10),
                     COL_NUMBER_IS_CONTAINED number (4) not null,
                     ID_IS_REFERENCED char(10),
                     ROW_NUMBER_IS_CONTAINED number (10) not null,
                     ID_IS_DISPLAYED char(10));
CREATE UNIQUE INDEX IDX31 ON ENTRY (ID);
CREATE TABLE EQUIP (ID char(10),
                     NUMBER1 number (10) not null);
CREATE UNIQUE INDEX IDX32 ON EQUIP (ID);
CREATE TABLE EQUIP_ALTEQIDS (ID_ALT_EQUIP char(10) not null,
                              ID_HAS_ALT char(10) not null);
CREATE UNIQUE INDEX IDX33 ON EQUIP_ALTEQIDS (ID_ALT_EQUIP,
                                              ID_HAS_ALT);
CREATE TABLE FAULT (ID char(10),
                     MTBF_TIME number (10),
                     ID_IS_DISPLAYED char(10));
CREATE UNIQUE INDEX IDX34 ON FAULT (ID);
CREATE TABLE FAULT_PARTINFO (ID_POINTS char(10) not null,
                              ID_IS_POINTED_TO char(10) not null);
CREATE UNIQUE INDEX IDX35 ON FAULT_PARTINFO (ID_POINTS,
                                              ID_IS_POINTED_TO);
CREATE TABLE FAULTINF_FAULT (ID_IS_IDENTIFIED char(10) not null,
                              ID_IDENTIFIES char(10) not null);
CREATE UNIQUE INDEX IDX36 ON FAULTINF_FAULT (ID_IS_IDENTIFIED,
                                              ID_IDENTIFIES);
CREATE TABLE FAULTINF_TEST (ID_IDENTIFIES char(10) not null,
                              ID_IS_IDENTIFIED char(10) not null);
CREATE UNIQUE INDEX IDX37 ON FAULTINF_TEST (ID_IDENTIFIES,
                                              ID_IS_IDENTIFIED);
CREATE TABLE FILLIN (ID char(10),

```

```

                                DEFAULT_DATA          char(80),
                                ID_IS_ASSERTED         char(10) not null,
                                RANGE_DATA             char(80),
                                ID_IS_DISPLAYED        char(10) not null);
CREATE UNIQUE INDEX IDX38 ON FILLIN (ID);
CREATE TABLE FLTSTATE (ID char(10),
                        ID_IS_IDENTIFIED char(10),
                        ID_IS_DISPLAYED char(10));
CREATE UNIQUE INDEX IDX39 ON FLTSTATE (ID);
CREATE TABLE FLTSTATE_EXPFAULT (ID_EXCULPATING char(10) not null,
                                ID_EXCULPATED char(10) not null);
CREATE UNIQUE INDEX IDX40 ON FLTSTATE_EXPFAULT (ID_EXCULPATING,
                                                ID_EXCULPATED);
CREATE TABLE FLTSTATE_IMPFAULT (ID_IMPLICATING char(10) not null,
                                ID_INPLICATED char(10) not null);
CREATE UNIQUE INDEX IDX41 ON FLTSTATE_IMPFAULT (ID_IMPLICATING,
                                                ID_INPLICATED);
CREATE TABLE FLTSTATE_WEIGHT (ID_SCALED char(10) not null,
                                WEIGHT_FACTOR_SCALES number (10) not null);
CREATE UNIQUE INDEX IDX42 ON FLTSTATE_WEIGHT (ID_SCALED,
                                                WEIGHT_FACTOR_SCALES);
CREATE TABLE GRAPHIC1 (ID char(10),
                        PENPATT_DATA_IS_SPECIFIES char(10),
                        PENSHAPE_DATA_IS_SPECIFIED char(10),
                        ID_IS_DISPALYED char(10));
CREATE UNIQUE INDEX IDX43 ON GRAPHIC1 (ID);
CREATE TABLE GRAPHIC_FOCUS (ID_IS_FOCUSED char(10) not null,
                              ID_SPECIFIES_FOCUS char(10) not null);
CREATE UNIQUE INDEX IDX44 ON GRAPHIC_FOCUS (ID_IS_FOCUSED,
                                              ID_SPECIFIES_FOCUS);
CREATE TABLE GRAPHIC_GRPGRIM (ID_INVOKES char(10) not null,
                                ID_IS_INVOKED char(10) not null);
CREATE UNIQUE INDEX IDX45 ON GRAPHIC_GRPGRIM (ID_INVOKES,
                                                ID_IS_INVOKED);
CREATE TABLE GRAPHIC_MINSIZE (ID_SPECIFIES char(10) not null,
                                MINSIZE_CODE_IS_SPECIFIES char(10) not null);
CREATE UNIQUE INDEX IDX46 ON GRAPHIC_MINSIZE (ID_SPECIFIES,
                                                MINSIZE_CODE_IS_SPECIFIES);
CREATE TABLE GRAPHIC_TRANSFORM (ID_SPECIFIES char(10) not null,
                                  TRANSF_DATA_IS_SPECIFIED number (9) not null);
CREATE UNIQUE INDEX IDX47 ON GRAPHIC_TRANSFORM (ID_SPECIFIES,
                                                  TRANSF_DATA_IS_SPECIFIED);
CREATE TABLE GRAPHIC_WINDOW (ID_SPECIFIES char(10) not null,
                                WINDOW_DATA_IS_SPECIFIED char(10) not null);
CREATE UNIQUE INDEX IDX48 ON GRAPHIC_WINDOW (ID_SPECIFIES,
                                              WINDOW_DATA_IS_SPECIFIED);
CREATE TABLE GRPHPRIM (ID char(10),
                        CODING_CODE_IS_SPECIFIED char(80),
                        FILE_DATA_IS_REFERENCED char(80),
                        PCTEXT_IS_CONTAINED long,
                        PENPATT_DATA_IS_SPECIFIED char(10),
                        PENSHAPE_DATA_IS_SPECIFIED char(10),

```

```

ID_IS_DISPLAYED char(10),
START_DATA char(15),
STOP_DATA char(15));
CREATE UNIQUE INDEX IDX49 ON GRPHPRIM (ID);
CREATE TABLE GRPHPRIM_MINSIZE (ID_SPECIFIES char(10) not null,
MINSIZE_CODE_IS_SPECIFIED char(10) not null);
CREATE UNIQUE INDEX IDX50 ON GRPHPRIM_MINSIZE (ID_SPECIFIES,
MINSIZE_CODE_IS_SPECIFIED);
CREATE TABLE GRPHPRIM_TRANSFORM (ID_SPECIFIES char(10) not null,
TRANSF_DATA_IS_SPECIFIED number (9) not null);
CREATE UNIQUE INDEX IDX51 ON GRPHPRIM_TRANSFORM (ID_SPECIFIES,
TRANSF_DATA_IS_SPECIFIED);
CREATE TABLE GRPHPRIM_WINDOW (ID_SPECIFIES char(10) not null,
WINDOW_DATA_IS_SPECIFIED char(10) not null);
CREATE UNIQUE INDEX IDX52 ON GRPHPRIM_WINDOW (ID_SPECIFIES,
WINDOW_DATA_IS_SPECIFIED);
CREATE TABLE LIST1 (ID char(10),
LIST_NAME char(20),
REFID char(10) not null,
ID_IS_CONTAINED char(10) not null);
CREATE UNIQUE INDEX IDX53 ON LIST1 (ID);
CREATE TABLE LIST_ELMNTREF (ID_IS_REFERENCED char(10) not null,
ID_REFERS char(10) not null);
CREATE UNIQUE INDEX IDX54 ON LIST_ELMNTREF (ID_IS_REFERENCED,
ID_REFERS);
CREATE TABLE MENU (ID char(10),
ID_IS_ASSERTED char(10) not null,
SELECT_DATA char(8),
ID_IS_DISPLAYED char(10) not null);
CREATE UNIQUE INDEX IDX55 ON MENU (ID);
CREATE TABLE MENU_CHOICE (ID char(10) not null,
ID_HAS char(10) not null);
CREATE UNIQUE INDEX IDX56 ON MENU_CHOICE (ID,
ID_HAS);
CREATE TABLE MENU_DEFAULT (DEFAULT_DATA char(80) not null,
ID_HAS char(10) not null);
CREATE UNIQUE INDEX IDX57 ON MENU_DEFAULT (DEFAULT_DATA,
ID_HAS);
CREATE TABLE NOTE (ID char(10),
ID_IS_DISPLAYED char(10) not null);
CREATE UNIQUE INDEX IDX58 ON NOTE (ID);
CREATE TABLE OPERINFO_DESCINFO (ID_IS_REFERENCED char(10) not null,
ID_REFERS char(10) not null);
CREATE UNIQUE INDEX IDX59 ON OPERINFO_DESCINFO (ID_IS_REFERENCED,
ID_REFERS);
CREATE TABLE OPERINFO_TASK (ID_REFERS char(10) not null,
ID_IS_REFERENCED char(10) not null);
CREATE UNIQUE INDEX IDX60 ON OPERINFO_TASK (ID_REFERS,
ID_IS_REFERENCED);
CREATE TABLE OUTCOME (ID char(10),
ID_IS_DISPLAYED char(10));
CREATE UNIQUE INDEX IDX61 ON OUTCOME (ID);

```

```

CREATE TABLE OUTCOME_FLTSTATE (ID_IS_IDENTIFIED char(10) not null,
                                ID_IDENTIFIES char(10) not null);
CREATE UNIQUE INDEX IDX62 ON OUTCOME_FLTSTATE (ID_IS_IDENTIFIED,
                                                ID_IDENTIFIES);
CREATE TABLE OUTCOME_PRECOND (ID_TESTS char(10) not null,
                                ID_IS_TESTED char(10) not null);
CREATE UNIQUE INDEX IDX63 ON OUTCOME_PRECOND (ID_TESTS,
                                              ID_IS_TESTED);
CREATE TABLE PARTBASE (ID char(10),
                        NSN char(20) not null,
                        ITEMID char(10),
                        NAME char(20),
                        PARTNUM char(80) not null,
                        REFID char(10) not null,
                        CAGE_CODE char(20) not null,
                        HCI_CODE char(10) not null,
                        SMR_CODE char(6) not null,
                        TYPE_DATA_IS_CONTAINED char(80));
CREATE UNIQUE INDEX IDX64 ON PARTBASE (ID);
CREATE UNIQUE INDEX IDX65 ON PARTBASE (NSN);
CREATE TABLE PARTBASE_XREF (ID_REFERS char(10) not null,
                             ID_IS_REFERENCED char(10) not null);
CREATE UNIQUE INDEX IDX66 ON PARTBASE_XREF (ID_REFERS,
                                             ID_IS_REFERENCED);
CREATE TABLE PARTINFO (ID char(10),
                        REFDES char(10) not null,
                        NOUNID char(80),
                        NOUNTYPE char(80),
                        INDX_NUMBER_REFERS char(15) not null,
                        MTBF_TIME number(10) not null,
                        REP_LEVEL_IS_IDENTIFIED char(10),
                        UNITSPER_NUMBER_IS_REQUIRED char(10) not null,
                        USABLON_CODE_IDENTIFIES char(15) not null);
CREATE UNIQUE INDEX IDX67 ON PARTINFO (ID);
CREATE UNIQUE INDEX IDX68 ON PARTINFO (REFDES);
CREATE TABLE PARTINFO_GRAPHIC (ID_IS_IDENTIFIED char(10) not null,
                                 ID_IDENTIFIES char(10) not null);
CREATE UNIQUE INDEX IDX69 ON PARTINFO_GRAPHIC (ID_IS_IDENTIFIED,
                                                ID_IDENTIFIES);
CREATE TABLE PARTINFO_PARTBASE (ID_IS_IDENTIFIED char(10) not null,
                                  ID_IDENTIFIES char(10) not null);
CREATE UNIQUE INDEX IDX70 ON PARTINFO_PARTBASE (ID_IS_IDENTIFIED,
                                                  ID_IDENTIFIES);
CREATE TABLE PERSON (ID char(10),
                      NUMBER1 number(10) not null);
CREATE UNIQUE INDEX IDX71 ON PERSON (ID);
CREATE TABLE PRECOND (ID char(10),
                       OP_DATA char(3),
                       POLARITY_DATA char(3),
                       ID_IS_TESTED char(10) not null);
CREATE UNIQUE INDEX IDX72 ON PRECOND (ID);
CREATE TABLE PRECOND_VALUE (ID_TESTS char(10) not null,

```

```

                                ID_IS_TESTED                char(10) not null);
CREATE UNIQUE INDEX IDX73 ON PRECOND_VALUE (ID_TESTS,
                                           ID_IS_TESTED);
CREATE TABLE PROCESS (ID char(10),
                       EXREF_ID_IS_REFERENCED char(10),
                       FILE_DATA_IS_REFERENCED char(80));
CREATE UNIQUE INDEX IDX74 ON PROCESS (ID);
CREATE TABLE PROMPT (ID char(10),
                      ID_IS_DISPLAYED char(10));
CREATE UNIQUE INDEX IDX75 ON PROMPT (ID);
CREATE TABLE PROMPT_FILLIN (ID char(10) not null,
                              ID_CONTAINS char(10) not null);
CREATE UNIQUE INDEX IDX76 ON PROMPT_FILLIN (ID,
                                             ID_CONTAINS);
CREATE TABLE PROMPT_MENU (ID_CALLED char(10) not null,
                           ID_CALLS char(10) not null);
CREATE UNIQUE INDEX IDX77 ON PROMPT_MENU (ID_CALLED,
                                           ID_CALLS);
CREATE TABLE PROPERTY (ID char(10),
                        ID_IS_REFERENCED char(10),
                        ID_IS_DISPLAYED char(10) not null);
CREATE UNIQUE INDEX IDX78 ON PROPERTY (ID);
CREATE TABLE RECT (ID char(10),
                    ACTION_CODE_IS_IDENTIFIED char(5),
                    AGENT_ID_PERFORMS char(7),
                    ID_IS_DISPLAYED char(10));
CREATE UNIQUE INDEX IDX79 ON RECT (ID);
CREATE TABLE RECT_FAULT (ID_IDENTIFIES char(10) not null,
                          ID_IS_IDENTIFIED char(10) not null);
CREATE UNIQUE INDEX IDX80 ON RECT_FAULT (ID_IDENTIFIES,
                                          ID_IS_IDENTIFIED);
CREATE TABLE RECT_TASK (ID_IDENTIFIES char(10) not null,
                          ID_IS_IDENTIFIED char(10) not null);
CREATE UNIQUE INDEX IDX81 ON RECT_TASK (ID_IDENTIFIES,
                                         ID_IS_IDENTIFIED);
CREATE TABLE RECT_TEST (ID_IDENTIFIES char(10) not null,
                          ID_IS_IDENTIFIED char(10) not null);
CREATE UNIQUE INDEX IDX82 ON RECT_TEST (ID_IDENTIFIES,
                                         ID_IS_IDENTIFIED);
CREATE TABLE REQCOND_ELMNTREF (ID_IS_REFERENCED char(10) not null,
                                ID_REFERS char(10) not null);
CREATE UNIQUE INDEX IDX83 ON REQCOND_ELMNTREF (ID_IS_REFERENCED,
                                                ID_REFERS);
CREATE TABLE REQCOND_PRECOND (ID_IS_TESTED char(10) not null,
                                ID_TESTS char(10) not null);
CREATE UNIQUE INDEX IDX84 ON REQCOND_PRECOND (ID_IS_TESTED,
                                              ID_TESTS);
CREATE TABLE STEP (ID char(10),
                    TIME_IS_REQUIRED number(10),
                    ID_IS_DISPLAYED char(10) not null);
CREATE UNIQUE INDEX IDX85 ON STEP (ID);
CREATE TABLE STEP_ANNOT (ID_IS_CONTAINED char(10) not null,

```


CREATE UNIQUE INDEX IDX86 ON STEP_ANNOT	ID_CONTAINS	char(10) not null);
	(ID_IS_CONTAINED,	
	ID_CONTAINS);	
CREATE TABLE STEP_ASSERTION	(ID_IS_ASSERTED	char(10) not null,
	ID_ASSERTS	char(10) not null);
CREATE UNIQUE INDEX IDX87 ON STEP_ASSERTION	(ID_IS_ASSERTED,	
	ID_ASSERTS);	
CREATE TABLE STEP_AUDIO	(ID_IS_INVOKED	char(10) not null,
	ID_INVOKES	char(10) not null);
CREATE UNIQUE INDEX IDX88 ON STEP_AUDIO	ID_IS_INVOKED,	
	ID_INVOKES);	
CREATE TABLE STEP_CAUTION	(ID_IS_REFERENCED	char(10) not null,
	ID_REFERS	char(10) not null);
CREATE UNIQUE INDEX IDX89 ON STEP_CAUTION	(ID_IS_REFERENCED,	
	ID_REFERS);	
CREATE TABLE STEP_COMSUM	(ID_IS_REQUIRED	char(10) not null,
	ID_REQUIRES	char(10) not null);
CREATE UNIQUE INDEX IDX90 ON STEP_COMSUM	(ID_IS_REQUIRED,	
	ID_REQUIRES);	
CREATE TABLE STEP_EQUIP	(ID_IS_REQUIRED	char(10) not null,
	ID_REQUIRES	char(10) not null);
CREATE UNIQUE INDEX IDX91 ON STEP_EQUIP	(ID_IS_REQUIRED,	
	ID_REQUIRES);	
CREATE TABLE STEP_GRAPHIC	(ID_IS_DISPLAYED	char(10) not null,
	ID_DISPLAYS	char(10) not null);
CREATE UNIQUE INDEX IDX92 ON STEP_GRAPHIC	(ID_IS_DISPLAYED,	
	ID_DISPLAYS);	
CREATE TABLE STEP_NOTE	(ID_IS_REFERENCED	char(10) not null,
	ID_REFERS	char(10) not null);
CREATE UNIQUE INDEX IDX93 ON STEP_NOTE	(ID_IS_REFERENCED,	
	ID_REFERS);	
CREATE TABLE STEP_PERSON	(ID_IS_REQUIRED	char(10) not null,
	ID_REQUIRES	char(10) not null);
CREATE UNIQUE INDEX IDX94 ON STEP_PERSON	(ID_IS_REQUIRED,	
	ID_REQUIRES);	
CREATE TABLE STEP_PROCESS	(ID_IS_INVOKED	char(10) not null,
	ID_INVOKES	char(10) not null);
CREATE UNIQUE INDEX IDX95 ON STEP_PROCESS	(ID_IS_INVOKED,	
	ID_INVOKES);	
CREATE TABLE STEP_PROMPT	(ID_IS_PRESENTED	char(10) not null,
	ID_PPRESENTS	char(10) not null);
CREATE UNIQUE INDEX IDX96 ON STEP_PROMPT	(ID_IS_PRESENTED,	
	ID_PPRESENTS);	
CREATE TABLE STEP_REQCOND	(ID_IS_TESTED	char(10) not null,
	ID_TESTS	char(10) not null);
CREATE UNIQUE INDEX IDX97 ON STEP_REQCOND	(ID_IS_TESTED,	
	ID_TESTS);	
CREATE TABLE STEP_SUBSTEP	(ID_IS_REFERENCED	char(10) not null,
	ID_REFERS	char(10) not null);
CREATE UNIQUE INDEX IDX98 ON STEP_SUBSTEP	(ID_IS_REFERENCED,	
	ID_REFERS);	
CREATE TABLE STEP_TABLE	(ID_PPRESENTS	char(10) not null,

```

CREATE TABLE STEP_TABLE (ID_IS_PRESENTED char(10) not null);
CREATE UNIQUE INDEX IDX99 ON STEP_TABLE (ID_PRESENTS, ID_IS_PRESENTED);
CREATE TABLE STEP_VERB (ID_USES char(10) not null, ID_IS_USED char(10) not null);
CREATE UNIQUE INDEX IDX100 ON STEP_VERB (ID_USES, ID_IS_USED);
CREATE TABLE STEP_VIDEO (ID_INVOKES char(10) not null, ID_IS_INVOKED char(10) not null);
CREATE UNIQUE INDEX IDX101 ON STEP_VIDEO (ID_INVOKES, ID_IS_INVOKED);
CREATE TABLE STEP_WARNING (ID_REFERS char(10) not null, ID_IS_REFERENCED char(10) not null);
CREATE UNIQUE INDEX IDX102 ON STEP_WARNING (ID_REFERS, ID_IS_REFERENCED);
CREATE TABLE SYSTEM_DESCINFO (ID_IS_REFERENCED char(10) not null, ID_REFERS char(10) not null);
CREATE UNIQUE INDEX IDX103 ON SYSTEM_DESCINFO (ID_IS_REFERENCED, ID_REFERS);
CREATE TABLE SYSTEM_FAULTINF (ID_IS_REFERENCED char(10) not null, ID_REFERS char(10) not null);
CREATE UNIQUE INDEX IDX104 ON SYSTEM_FAULTINF (ID_IS_REFERENCED, ID_REFERS);
CREATE TABLE SYSTEM_OPERINFO (ID_IS_REFERENCED char(10) not null, ID_REFERS char(10) not null);
CREATE UNIQUE INDEX IDX105 ON SYSTEM_OPERINFO (ID_IS_REFERENCED, ID_REFERS);
CREATE TABLE SYSTEM_PARTINFO (ID_IS_IDENTIFIED char(10) not null, ID_IDENTIFIES char(10) not null);
CREATE UNIQUE INDEX IDX106 ON SYSTEM_PARTINFO (ID_IS_IDENTIFIED, ID_IDENTIFIES);
CREATE TABLE SYSTEM_SUBSYSTEM (ID_IS_REFERENCED char(10) not null, ID_REFERS char(10) not null);
CREATE UNIQUE INDEX IDX107 ON SYSTEM_SUBSYSTEM (ID_IS_REFERENCED, ID_REFERS);
CREATE TABLE SYSTEM_TASK (ID_REFERS char(10) not null, ID_IS_REFERENCED char(10) not null);
CREATE UNIQUE INDEX IDX108 ON SYSTEM_TASK (ID_REFERS, ID_IS_REFERENCED);
CREATE TABLE TABLE_COLHDDEF (ID_IS_CONTAINED char(10) not null, ID_CONTAINS char(10) not null);
CREATE UNIQUE INDEX IDX109 ON TABLE_COLHDDEF (ID_IS_CONTAINED, ID_CONTAINS);
CREATE TABLE TABLE_ENTRY (ID_IS_CONTAINED char(10) not null, ID_CONTAINS char(10) not null);
CREATE UNIQUE INDEX IDX110 ON TABLE_ENTRY (ID_IS_CONTAINED, ID_CONTAINS);
CREATE TABLE TASK (ID char(10), TIME_IS_REQUIRED number(10));
CREATE UNIQUE INDEX IDX111 ON TASK (ID);
CREATE TABLE TASK_CAUTION (ID_IS_REFERENCED char(10) not null, ID_REFERS char(10) not null);

```

```

CREATE UNIQUE INDEX IDX112 ON TASK_CAUTION      (ID_IS_REFERENCED,
                                                    ID_REFERS);
CREATE TABLE TASK_CONSUM      (ID_IS_REQUIRED      char(10) not null,
                                ID_REQUIRES          char(10) not null);
CREATE UNIQUE INDEX IDX113 ON TASK_CONSUM      (ID_IS_REQUIRED,
                                                    ID_REQUIRES);
CREATE TABLE TASK_EQUIP      (ID_IS_REQUIRED      char(10) not null,
                                ID_REQUIRES          char(10) not null);
CREATE UNIQUE INDEX IDX114 ON TASK_EQUIP      (ID_IS_REQUIRED,
                                                    ID_REQUIRES);
CREATE TABLE TASK_FOLLOWON    (ID_FOLLOWED          char(10) not null,
                                REFID_FOLLOWING       char(10) not null);
CREATE UNIQUE INDEX IDX115 ON TASK_FOLLOWON    (ID_FOLLOWED,
                                                    REFID_FOLLOWING);
CREATE TABLE TASK_NOTE      (ID_IS_REFERENCED      char(10) not null,
                                ID_REFERS             char(10) not null);
CREATE UNIQUE INDEX IDX116 ON TASK_NOTE      (ID_IS_REFERENCED,
                                                    ID_REFERS);
CREATE TABLE TASK_PERSON      (ID_IS_REQUIRED      char(10) not null,
                                ID_REQUIRES          char(10) not null);
CREATE UNIQUE INDEX IDX117 ON TASK_PERSON      (ID_IS_REQUIRED,
                                                    ID_REQUIRES);
CREATE TABLE TASK_REQCOND     (ID_IS_TESTED        char(10) not null,
                                ID_TESTS              char(10) not null);
CREATE UNIQUE INDEX IDX118 ON TASK_REQCOND     (ID_IS_TESTED,
                                                    ID_TESTS);
CREATE TABLE TASK_STEP      (ID_IS_REFERENCED      char(10) not null,
                                ID_REFERS             char(10) not null);
CREATE UNIQUE INDEX IDX119 ON TASK_STEP      (ID_IS_REFERENCED,
                                                    ID_REFERS);
CREATE TABLE TASK_VERB      (ID_USES              char(10) not null,
                                ID_IS_USED            char(10) not null);
CREATE UNIQUE INDEX IDX120 ON TASK_VERB      (ID_USES,
                                                    ID_IS_USED);
CREATE TABLE TASK_WARNING    (ID_REFERS            char(10) not null,
                                ID_IS_REFERENCED       char(10) not null);
CREATE UNIQUE INDEX IDX121 ON TASK_WARNING    (ID_REFERS,
                                                    ID_IS_REFERENCED);
CREATE TABLE TECHINFO_SYSTEM (ID_IS_CONTAINED      char(10) not null,
                                ID_CONTAINS            char(10) not null);
CREATE UNIQUE INDEX IDX122 ON TECHINFO_SYSTEM (ID_IS_CONTAINED,
                                                    ID_CONTAINS);
CREATE TABLE TEST            (ID                    char(10),
                                AGENT_ID_PERFORMS      char(7),
                                RANGE_DATA_REPRESENTS char(80),
                                ID_IS_DISPLAYED        char(10));
CREATE UNIQUE INDEX IDX123 ON TEST            (ID);
CREATE TABLE TEST_OUTCOME    (ID_IS_PRODUCED       char(10) not null,
                                ID_PRODUCES           char(10) not null);
CREATE UNIQUE INDEX IDX124 ON TEST_OUTCOME    (ID_IS_PRODUCED,
                                                    ID_PRODUCES);
CREATE TABLE TEST_TASK      (ID_IS_IDENTIFIED      char(10) not null,

```

```

CREATE UNIQUE INDEX IDX125 ON TEST_TASK ID_IDENTIFIES char(10) not null);
(ID_IS_IDENTIFIED,
ID_IDENTIFIES);
CREATE TABLE TEXT (ID char(10),
ATT_VALUE_IS_CONTAINED char(10),
PCTEXT_IS_CONTAINED long);
CREATE UNIQUE INDEX IDX126 ON TEXT (ID);
CREATE TABLE VALUE (ID char(10),
ID_IS_DISPLAYED char(10) not null);
CREATE UNIQUE INDEX IDX127 ON VALUE (ID);
CREATE TABLE VIDEO (ID char(10),
EXREF_ID_IS_REFERENCED char(10),
FILE_DATA_IS_REFERENCED char(80));
CREATE UNIQUE INDEX IDX128 ON VIDEO (ID);
CREATE TABLE WARNING (ID char(10),
ID_IS_DISPLAYED char(10) not null);
CREATE UNIQUE INDEX IDX129 ON WARNING (ID);
CREATE TABLE XREF (ID char(10),
ID_IS_REFERENCED char(10),
EXREF_ID_IS_REFERENCED char(10),
RELATION_DATA_IS_CONTAINED char(10));
CREATE UNIQUE INDEX IDX130 ON XREF (ID);
CREATE TABLE XREF_ATTNAME (ID_HAS char(10) not null,
ATTNAME char(15) not null);
CREATE UNIQUE INDEX IDX131 ON XREF_ATTNAME (ID_HAS);

```

THIS PAGE INTENTIONALLY LEFT BLANK

80.2 Constraints Not Used for ORACLE Database Schema Generation.

CONSTRAINT NAME : CDM-EXCLUDES

FIELD ID OF ASSERTION
AND
FIELD ID OF CHOICE
AND
FIELD ID OF COLHDDEF
AND
FIELD ID OF COMMON_ELEMENT
AND
FIELD ID OF CONTEXT
AND
FIELD ID OF ENTRY
AND
FIELD ID OF LIST1
AND
FIELD ID OF PARTBASE
AND
FIELD ID OF PRECOND
AND
FIELD ID OF PROPERTY
AND
FIELD ID OF VALUE
AND
FIELD ID OF XREF
ARE MUTUALLY EXCLUSIVE

CONSTRAINT NAME : COMMON-ELEM-EXCL

FIELD ID OF ANNOT
AND
FIELD ID OF AUDIO
AND
FIELD ID OF CAUTION
AND
FIELD ID OF CONSUM
AND
FIELD ID OF DESCINFO
AND
FIELD ID_REFERS OF DICTITEM_ELMNTREF
AND
FIELD ID OF EQUIP
AND
FIELD ID OF FAULT
AND
FIELD ID_IDENTIFIES OF FAULTINF_FAULT
FIELD ID_IDENTIFIES OF FAULTINF_TEST
FIELD ID_IS_REFERENCED OF SYSTEM_FAULTINF
AND
FIELD ID OF FILLIN
AND
FIELD ID OF FLTSTATE

AND
 FIELD ID OF GRAPHIC1
 AND
 FIELD ID OF GRPHPRIM
 AND
 FIELD ID OF MENU
 AND
 FIELD ID OF NOTE
 AND
 FIELD ID_REFERS OF OPERINFO_DESCINFO
 FIELD ID_REFERS OF OPERINFO_TASK
 FIELD ID_IS_REFERENCED OF SYSTEM_OPERINFO
 AND
 FIELD ID OF OUTCOME
 AND
 FIELD ID OF PARTINFO
 AND
 FIELD ID OF PERSON
 AND
 FIELD ID OF PROCESS
 AND
 FIELD ID OF PROMPT
 AND
 FIELD ID OF RECT
 AND
 FIELD ID_REFERS OF REQCOND_ELMNTREF
 FIELD ID_TESTS OF REQCOND_PRECOND
 FIELD ID_JS_TESTED OF STEP_REQCOND
 FIELD ID_IS_TESTED OF TASK_REQCOND
 AND
 FIELD ID OF STEP
 AND
 FIELD ID_REFERS OF SYSTEM_DESCINFO
 FIELD ID_REFERS OF SYSTEM_FAULTINF
 FIELD ID_REFERS OF SYSTEM_OPERINFO
 FIELD ID_IDENTIFIES OF SYSTEM_PARTINFO
 FIELD ID_IS_REFERENCED OF SYSTEM_SUBSYSTEM
 FIELD ID_REFERS OF SYSTEM_SUBSYSTEM
 FIELD ID_REFERS OF SYSTEM_TASK
 FIELD ID_IS_CONTAINED OF TECHINFO_SYSTEM
 AND
 FIELD ID_IS_PRESENTED OF DESCINFO_TABLE
 FIELD ID_IS_PRESENTED OF STEP_TABLE
 FIELD ID_CONTAINS OF TABLE_COLHDDEF
 FIELD ID_CONTAINS OF TABLE_ENTRY
 AND
 FIELD ID OF TASK
 AND
 FIELD ID_CONTAINS OF TECHINFO_SYSTEM
 AND
 FIELD ID OF TEST
 AND

FIELD ID OF TEXT
AND
FIELD ID_IS_USED OF STEP_VERB
FIELD ID_IS_USED OF TASK_VERB
AND
FIELD ID OF VIDEO
AND
FIELD ID OF WARNING
ARE MUTUALLY EXCLUSIVE

CONSTRAINT NAME : FAULT-FLTSTATE
FIELD ID_EXCULPATED OF FLTSTATE_EXPFAULT
FIELD ID_EXCULPATING OF FLTSTATE_EXPFAULT
AND
FIELD ID_INPLICATED OF FLTSTATE_IMPFAULT
FIELD ID_IMPLICATING OF FLTSTATE_IMPFAULT
ARE MUTUALLY EXCLUSIVE

CONSTRAINT NAME : sub001
FIELD ID OF ANNOT
IS A SUBSET OF
FIELD ID OF COMMON_ELEMENT

CONSTRAINT NAME : sub002
FIELD ID_IS_DISPLAYED OF ANNOT
IS A SUBSET OF
FIELD ID OF TEXT

CONSTRAINT NAME : sub003
FIELD ID_IS_ASSERTED OF ASSERTION
IS A SUBSET OF
FIELD ID OF PROPERTY

CONSTRAINT NAME : sub004
FIELD ID_ASSERTS OF ASSERTION__VALUE
IS A SUBSET OF
FIELD ID OF ASSERTION

CONSTRAINT NAME : sub005
FIELD ID_IS_ASSERTED OF ASSERTION__VALUE
IS A SUBSET OF
FIELD ID OF VALUE

CONSTRAINT NAME : sub006
FIELD ID OF AUDIO
IS A SUBSET OF
FIELD ID OF COMMON_ELEMENT

CONSTRAINT NAME : sub007
FIELD ID OF CAUTION
IS A SUBSET OF
FIELD ID OF COMMON_ELEMENT

CONSTRAINT NAME : sub008
 FIELD ID_IS_DISPLAYED OF CAUTION
 IS A SUBSET OF
 FIELD ID OF TEXT

CONSTRAINT NAME : sub009
 FIELD ID_IS_DISPLAYED OF CHOICE
 IS A SUBSET OF
 FIELD ID OF TEXT

CONSTRAINT NAME : sub010
 FIELD ID_CONTAINS OF CHOICE_VALUE
 IS A SUBSET OF
 FIELD ID OF CHOICE

CONSTRAINT NAME : sub011
 FIELD ID_IS_CONTAINED OF CHOICE_VALUE
 IS A SUBSET OF
 FIELD ID OF VALUE

CONSTRAINT NAME : sub012
 FIELD ID_REFERS OF COMMON_ELEM_XREF
 IS A SUBSET OF
 FIELD ID OF COMMON_ELEMENT

CONSTRAINT NAME : sub013
 FIELD ID_IS_REFERENCED OF COMMON_ELEM_XREF
 IS A SUBSET OF
 FIELD ID OF XREF

CONSTRAINT NAME : sub014
 FIELD ID_IS_CONTAINED OF COMMON_ELEMENT
 IS A SUBSET OF
 FIELD ID OF CONTEXT

CONSTRAINT NAME : sub015
 FIELD ID OF CONSUM
 IS A SUBSET OF
 FIELD ID OF COMMON_ELEMENT

CONSTRAINT NAME : sub016
 FIELD ID_CONTAINS OF CONTEXT_CONFIG
 IS A SUBSET OF
 FIELD ID OF CONTEXT

CONSTRAINT NAME : sub017
 FIELD ID_TESTS OF CONTEXT_PRECOND
 IS A SUBSET OF
 FIELD ID OF CONTEXT

CONSTRAINT NAME : sub018

FIELD ID_IS_TESTED OF CONTEXT_PRECOND
 IS A SUBSET OF
 FIELD ID OF PRECOND

CONSTRAINT NAME : sub019
 FIELD ID_CONTAINS OF CONTEXT_RELEASE
 IS A SUBSET OF
 FIELD ID OF CONTEXT

CONSTRAINT NAME : sub020
 FIELD ID_CONTAINS OF CONTEXT_TRACK
 IS A SUBSET OF
 FIELD ID OF CONTEXT

CONSTRAINT NAME : sub021
 FIELD ID_CONTAINS OF CONTEXT_VERSION
 IS A SUBSET OF
 FIELD ID OF CONTEXT

CONSTRAINT NAME : sub022
 FIELD ID OF DESCINFO
 IS A SUBSET OF
 FIELD ID OF COMMON_ELEMENT

CONSTRAINT NAME : sub023
 FIELD ID_IS_DISPLAYED OF DESCINFO
 IS A SUBSET OF
 FIELD ID OF TEXT

CONSTRAINT NAME : sub024
 FIELD ID_IS_REFERENCED OF DESCINFO_ANNOT
 IS A SUBSET OF
 FIELD ID OF ANNOT

CONSTRAINT NAME : sub025
 FIELD ID_REFERS OF DESCINFO_ANNOT
 IS A SUBSET OF
 FIELD ID OF DESCINFO

CONSTRAINT NAME : sub026
 FIELD ID_IS_ASSERTED OF DESCINFO_ASSERTION
 IS A SUBSET OF
 FIELD ID OF ASSERTION

CONSTRAINT NAME : sub027
 FIELD ID_ASSERTS OF DESCINFO_ASSERTION
 IS A SUBSET OF
 FIELD ID OF DESCINFO

CONSTRAINT NAME : sub028
 FIELD ID_IS_INVOKED OF DESCINFO_AUDIO
 IS A SUBSET OF

FIELD ID OF AUDIO

CONSTRAINT NAME : sub029
 FIELD ID_INVOKES OF DESCINFO_AUDIO
 IS A SUBSET OF
 FIELD ID OF DESCINFO

CONSTRAINT NAME : sub030
 FIELD ID_IS REFERENCED OF DESCINFO_DESCINFO
 IS A SUBSET OF
 FIELD ID OF DESCINFO

CONSTRAINT NAME : sub031
 FIELD ID_REFERS OF DESCINFO_DESCINFO
 IS A SUBSET OF
 FIELD ID OF DESCINFO

CONSTRAINT NAME : sub032
 FIELD ID_DISPLAYS OF DESCINFO_GRAPHIC
 IS A SUBSET OF
 FIELD ID OF DESCINFO

CONSTRAINT NAME : sub033
 FIELD ID_IS_DISPLAYED OF DESCINFO_GRAPHIC
 IS A SUBSET OF
 FIELD ID OF GRAPHIC1

CONSTRAINT NAME : sub034
 FIELD ID_INVOKES OF DESCINFO_PROCESS
 IS A SUBSET OF
 FIELD ID OF DESCINFO

CONSTRAINT NAME : sub035
 FIELD ID_IS_INVOKED OF DESCINFO_PROCESS
 IS A SUBSET OF
 FIELD ID OF PROCESS

CONSTRAINT NAME : sub036
 FIELD ID_PPRESENTS OF DESCINFO_PROMPT
 IS A SUBSET OF
 FIELD ID OF DESCINFO

CONSTRAINT NAME : sub037
 FIELD ID_IS_PRESENTED OF DESCINFO_PROMPT
 IS A SUBSET OF
 FIELD ID OF PROMPT

CONSTRAINT NAME : sub038
 FIELD ID_PPRESENTS OF DESCINFO_TABLE
 IS A SUBSET OF
 FIELD ID OF DESCINFO

CONSTRAINT NAME : sub039
FIELD ID_IS_PRESENTED OF DESCINFO_TABLE
IS A SUBSET OF
FIELD ID_CONTAINS OF TABLE_COLHDDEF

CONSTRAINT NAME : sub040
FIELD ID_IS_PRESENTED OF DESCINFO_TABLE
IS A SUBSET OF
FIELD ID_CONTAINS OF TABLE_ENTRY

CONSTRAINT NAME : sub041
FIELD ID_INVOKES OF DESCINFO_VIDEO
IS A SUBSET OF
FIELD ID OF DESCINFO

CONSTRAINT NAME : sub042
FIELD ID_IS_INVOKED OF DESCINFO_VIDEO
IS A SUBSET OF
FIELD ID OF VIDEO

CONSTRAINT NAME : sub043
FIELD ID_REFERS OF DICTITEM_ELMNTREF
IS A SUBSET OF
FIELD ID OF COMMON_ELEMENT

CONSTRAINT NAME : sub044
FIELD ID_IS_DISPLAYED OF ENTRY
IS A SUBSET OF
FIELD ID OF TEXT

CONSTRAINT NAME : sub045
FIELD ID OF EQUIP
IS A SUBSET OF
FIELD ID OF COMMON_ELEMENT

CONSTRAINT NAME : sub046
FIELD ID_ALT_EQUIP OF EQUIP_ALTEQIDS
IS A SUBSET OF
FIELD ID OF EQUIP

CONSTRAINT NAME : sub047
FIELD ID_HAS_ALT OF EQUIP_ALTEQIDS
IS A SUBSET OF
FIELD ID OF EQUIP

CONSTRAINT NAME : sub048
FIELD ID OF FAULT
IS A SUBSET OF
FIELD ID OF COMMON_ELEMENT

CONSTRAINT NAME : sub049
FIELD ID_IS_DISPLAYED OF FAULT

IS A SUBSET OF
 FIELD ID OF TEXT

CONSTRAINT NAME : sub050
 FIELD ID_POINTS OF FAULT_PARTINFO
 IS A SUBSET OF
 FIELD ID OF FAULT

CONSTRAINT NAME : sub051
 FIELD ID_IS POINTED TO OF FAULT_PARTINFO
 IS A SUBSET OF
 FIELD ID OF PARTINFO

CONSTRAINT NAME : sub052
 FIELD ID_IDENTIFIES OF FAULTINF_FAULT
 IS A SUBSET OF
 FIELD ID OF COMMON_ELEMENT

CONSTRAINT NAME : sub053
 FIELD ID_IDENTIFIES OF FAULTINF_TEST
 IS A SUBSET OF
 FIELD ID OF COMMON_ELEMENT

CONSTRAINT NAME : sub054
 FIELD ID_IS IDENTIFIED OF FAULTINF_FAULT
 IS A SUBSET OF
 FIELD ID OF FAULT

CONSTRAINT NAME : sub055
 FIELD ID_IS IDENTIFIED OF FAULTINF_TEST
 IS A SUBSET OF
 FIELD ID OF TEST

CONSTRAINT NAME : sub056
 FIELD ID OF FILLIN
 IS A SUBSET OF
 FIELD ID OF COMMON_ELEMENT

CONSTRAINT NAME : sub057
 FIELD ID_IS ASSERTED OF FILLIN
 IS A SUBSET OF
 FIELD ID OF PROPERTY

CONSTRAINT NAME : sub058
 FIELD ID_IS DISPLAYED OF FILLIN
 IS A SUBSET OF
 FIELD ID OF TEXT

CONSTRAINT NAME : sub059
 FIELD ID OF FLTSTATE
 IS A SUBSET OF
 FIELD ID OF COMMON_ELEMENT

CONSTRAINT NAME : sub060
FIELD ID_IS_IDENTIFIED OF FLTSTATE
IS A SUBSET OF
FIELD ID OF TEST

CONSTRAINT NAME : sub061
FIELD ID_IS_DISPLAYED OF FLTSTATE
IS A SUBSET OF
FIELD ID OF TEXT

CONSTRAINT NAME : sub062
FIELD ID_EXCULPATING OF FLTSTATE_EXPFAULT
IS A SUBSET OF
FIELD ID OF FAULT

CONSTRAINT NAME : sub063
FIELD ID_EXCULPATED OF FLTSTATE_EXPFAULT
IS A SUBSET OF
FIELD ID OF FLTSTATE

CONSTRAINT NAME : sub064
FIELD ID_IMPLICATING OF FLTSTATE_IMPFAULT
IS A SUBSET OF
FIELD ID OF FAULT

CONSTRAINT NAME : sub065
FIELD ID_INPLICATED OF FLTSTATE_IMPFAULT
IS A SUBSET OF
FIELD ID OF FLTSTATE

CONSTRAINT NAME : sub066
FIELD ID_SCALED OF FLTSTATE_WEIGHT
IS A SUBSET OF
FIELD ID OF FLTSTATE

CONSTRAINT NAME : sub067
FIELD ID OF GRAPHIC1
IS A SUBSET OF
FIELD ID OF COMMON_ELEMENT

CONSTRAINT NAME : sub068
FIELD ID_IS_DISPALYED OF GRAPHIC1
IS A SUBSET OF
FIELD ID OF TEXT

CONSTRAINT NAME : sub069
FIELD ID_IS_FOCUSED OF GRAPHIC_FOCUS
IS A SUBSET OF
FIELD ID OF GRAPHIC1

CONSTRAINT NAME : sub070

FIELD ID_SPECIFIES_FOCUS OF GRAPHIC_FOCUS
 IS A SUBSET OF
 FIELD ID OF GRAPHIC1

CONSTRAINT NAME : sub071
 FIELD ID_INVOKES OF GRAPHIC_GRPGRIM
 IS A SUBSET OF
 FIELD ID OF GRAPHIC1

CONSTRAINT NAME : sub072
 FIELD ID_IS_INVOKED OF GRAPHIC_GRPGRIM
 IS A SUBSET OF
 FIELD ID OF GRPHPRIM

CONSTRAINT NAME : sub073
 FIELD ID_SPECIFIES OF GRAPHIC_MINSIZE
 IS A SUBSET OF
 FIELD ID OF GRAPHIC1

CONSTRAINT NAME : sub074
 FIELD ID_SPECIFIES OF GRAPHIC_TRANSFORM
 IS A SUBSET OF
 FIELD ID OF GRAPHIC1

CONSTRAINT NAME : sub075
 FIELD ID_SPECIFIES OF GRAPHIC_WINDOW
 IS A SUBSET OF
 FIELD ID OF GRAPHIC1

CONSTRAINT NAME : sub076
 FIELD ID OF GRPHPRIM
 IS A SUBSET OF
 FIELD ID OF COMMON_ELEMENT

CONSTRAINT NAME : sub077
 FIELD ID_IS_DISPLAYED OF GRPHPRIM
 IS A SUBSET OF
 FIELD ID OF TEXT

CONSTRAINT NAME : sub078
 FIELD ID_SPECIFIES OF GRPHPRIM_MINSIZE
 IS A SUBSET OF
 FIELD ID OF GRPHPRIM

CONSTRAINT NAME : sub079
 FIELD ID_SPECIFIES OF GRPHPRIM_TRANSFORM
 IS A SUBSET OF
 FIELD ID OF GRPHPRIM

CONSTRAINT NAME : sub080
 FIELD ID_SPECIFIES OF GRPHPRIM_WINDOW
 IS A SUBSET OF

FIELD ID OF GRPHPRIM

CONSTRAINT NAME : sub081
FIELD ID_IS_CONTAINED OF LIST1
IS A SUBSET OF
FIELD ID OF CONTEXT

CONSTRAINT NAME : sub082
FIELD ID_REFERS OF LIST_ELMNTREF
IS A SUBSET OF
FIELD ID OF LIST1

CONSTRAINT NAME : sub083
FIELD ID OF MENU
IS A SUBSET OF
FIELD ID OF COMMON_ELEMENT

CONSTRAINT NAME : sub084
FIELD ID_IS_ASSERTED OF MENU
IS A SUBSET OF
FIELD ID OF PROPERTY

CONSTRAINT NAME : sub085
FIELD ID_IS_DISPLAYED OF MENU
IS A SUBSET OF
FIELD ID OF TEXT

CONSTRAINT NAME : sub086
FIELD ID OF MENU_CHOICE
IS A SUBSET OF
FIELD ID OF CHOICE

CONSTRAINT NAME : sub087
FIELD ID_HAS OF MENU_CHOICE
IS A SUBSET OF
FIELD ID OF MENU

CONSTRAINT NAME : sub088
FIELD ID_HAS OF MENU_DEFAULT
IS A SUBSET OF
FIELD ID OF MENU

CONSTRAINT NAME : sub089
FIELD ID OF NOTE
IS A SUBSET OF
FIELD ID OF COMMON_ELEMENT

CONSTRAINT NAME : sub090
FIELD ID_IS_DISPLAYED OF NOTE
IS A SUBSET OF
FIELD ID OF TEXT

CONSTRAINT NAME : sub091
FIELD ID_IS_REFERENCED OF OPERINFO_DESCINFO
IS A SUBSET OF
FIELD ID OF DESCINFO

CONSTRAINT NAME : sub092
FIELD ID_IS_REFERENCED OF OPERINFO_TASK
IS A SUBSET OF
FIELD ID OF TASK

CONSTRAINT NAME : sub093
FIELD ID OF OUTCOME
IS A SUBSET OF
FIELD ID OF COMMON_ELEMENT

CONSTRAINT NAME : sub094
FIELD ID_IS_DISPLAYED OF OUTCOME
IS A SUBSET OF
FIELD ID OF TEXT

CONSTRAINT NAME : sub095
FIELD ID_IS_IDENTIFIED OF OUTCOME_FLTSTATE
IS A SUBSET OF
FIELD ID OF FLTSTATE

CONSTRAINT NAME : sub096
FIELD ID_IDENTIFIES OF OUTCOME_FLTSTATE
IS A SUBSET OF
FIELD ID OF OUTCOME

CONSTRAINT NAME : sub097
FIELD ID_TESTS OF OUTCOME_PRECOND
IS A SUBSET OF
FIELD ID OF OUTCOME

CONSTRAINT NAME : sub098
FIELD ID_IS_TESTED OF OUTCOME_PRECOND
IS A SUBSET OF
FIELD ID OF PRECOND

CONSTRAINT NAME : sub099
FIELD ID_REFERS OF PARTBASE_XREF
IS A SUBSET OF
FIELD ID OF PARTBASE

CONSTRAINT NAME : sub100
FIELD ID_IS_REFERENCED OF PARTBASE_XREF
IS A SUBSET OF
FIELD ID OF XREF

CONSTRAINT NAME : sub101
FIELD ID OF PARTINFO

IS A SUBSET OF
FIELD ID OF COMMON_ELEMENT

CONSTRAINT NAME : sub102
FIELD ID_IS_IDENTIFIED OF PARTINFO_GRAPHIC
IS A SUBSET OF
FIELD ID OF GRAPHIC1

CONSTRAINT NAME : sub103
FIELD ID_IDENTIFIES OF PARTINFO_GRAPHIC
IS A SUBSET OF
FIELD ID OF PARTINFO

CONSTRAINT NAME : sub104
FIELD ID_IS_IDENTIFIED OF PARTINFO_PARTBASE
IS A SUBSET OF
FIELD ID OF PARTBASE

CONSTRAINT NAME : sub105
FIELD ID_IDENTIFIES OF PARTINFO_PARTBASE
IS A SUBSET OF
FIELD ID OF PARTINFO

CONSTRAINT NAME : sub106
FIELD ID OF PERSON
IS A SUBSET OF
FIELD ID OF COMMON_ELEMENT

CONSTRAINT NAME : sub107
FIELD ID_IS_TESTED OF PRECOND
IS A SUBSET OF
FIELD ID OF PROPERTY

CONSTRAINT NAME : sub108
FIELD ID_TESTS OF PRECOND_VALUE
IS A SUBSET OF
FIELD ID OF PRECOND

CONSTRAINT NAME : sub109
FIELD ID_IS_TESTED OF PRECOND_VALUE
IS A SUBSET OF
FIELD ID OF VALUE

CONSTRAINT NAME : sub110
FIELD ID OF PROCESS
IS A SUBSET OF
FIELD ID OF COMMON_ELEMENT

CONSTRAINT NAME : sub111
FIELD ID OF PROMPT
IS A SUBSET OF
FIELD ID OF COMMON_ELEMENT

CONSTRAINT NAME : sub112
 FIELD ID_IS_DISPLAYED OF PROMPT
 IS A SUBSET OF
 FIELD ID OF TEXT

CONSTRAINT NAME : sub113
 FIELD ID OF PROMPT_FILLIN
 IS A SUBSET OF
 FIELD ID OF FILLIN

CONSTRAINT NAME : sub114
 FIELD ID_CONTAINS OF PROMPT_FILLIN
 IS A SUBSET OF
 FIELD ID OF PROMPT

CONSTRAINT NAME : sub115
 FIELD ID_CALLED OF PROMPT_MENU
 IS A SUBSET OF
 FIELD ID OF MENU

CONSTRAINT NAME : sub116
 FIELD ID_CALLS OF PROMPT_MENU
 IS A SUBSET OF
 FIELD ID OF PROMPT

CONSTRAINT NAME : sub117
 FIELD ID_IS_DISPLAYED OF PROPERTY
 IS A SUBSET OF
 FIELD ID OF TEXT

CONSTRAINT NAME : sub118
 FIELD ID OF RECT
 IS A SUBSET OF
 FIELD ID OF COMMON_ELEMENT

CONSTRAINT NAME : sub119
 FIELD ID_IS_DISPLAYED OF RECT
 IS A SUBSET OF
 FIELD ID OF TEXT

CONSTRAINT NAME : sub120
 FIELD ID_IDENTIFIES OF RECT_FAULT
 IS A SUBSET OF
 FIELD ID OF FAULT

CONSTRAINT NAME : sub121
 FIELD ID_IS_IDENTIFIED OF RECT_FAULT
 IS A SUBSET OF
 FIELD ID OF RECT

CONSTRAINT NAME : sub122

FIELD ID_IDENTIFIES OF RECT_TASK
IS A SUBSET OF
FIELD ID OF RECT

CONSTRAINT NAME : sub123
FIELD ID_IS_IDENTIFIED OF RECT_TASK
IS A SUBSET OF
FIELD ID OF TASK

CONSTRAINT NAME : sub124
FIELD ID_IDENTIFIES OF RECT_TEST
IS A SUBSET OF
FIELD ID OF RECT

CONSTRAINT NAME : sub125
FIELD ID_IS_IDENTIFIED OF RECT_TEST
IS A SUBSET OF
FIELD ID OF TEST

CONSTRAINT NAME : sub126
FIELD ID_IS_TESTED OF REQCOND_PRECOND
IS A SUBSET OF
FIELD ID OF PRECOND

CONSTRAINT NAME : sub127
FIELD ID OF STEP
IS A SUBSET OF
FIELD ID OF COMMON_ELEMENT

CONSTRAINT NAME : sub128
FIELD ID_IS_DISPLAYED OF STEP
IS A SUBSET OF
FIELD ID OF TEXT

CONSTRAINT NAME : sub129
FIELD ID_IS_CONTAINED OF STEP_ANNOT
IS A SUBSET OF
FIELD ID OF ANNOT

CONSTRAINT NAME : sub130
FIELD ID_CONTAINS OF STEP_ANNOT
IS A SUBSET OF
FIELD ID OF STEP

CONSTRAINT NAME : sub131
FIELD ID_IS_ASSERTED OF STEP_ASSERTION
IS A SUBSET OF
FIELD ID OF ASSERTION

CONSTRAINT NAME : sub132
FIELD ID_ASSERTS OF STEP_ASSERTION
IS A SUBSET OF

FIELD ID OF STEP

CONSTRAINT NAME : sub133
FIELD ID_IS_INVOKED OF STEP_AUDIO
IS A SUBSET OF
FIELD ID OF AUDIO

CONSTRAINT NAME : sub134
FIELD ID_INVOKES OF STEP_AUDIO
IS A SUBSET OF
FIELD ID OF STEP

CONSTRAINT NAME : sub135
FIELD ID_IS_REFERENCED OF STEP_CAUTION
IS A SUBSET OF
FIELD ID OF CAUTION

CONSTRAINT NAME : sub136
FIELD ID_REFERS OF STEP_CAUTION
IS A SUBSET OF
FIELD ID OF STEP

CONSTRAINT NAME : sub137
FIELD ID_IS_REQUIRED OF STEP_COMSUM
IS A SUBSET OF
FIELD ID OF CONSUM

CONSTRAINT NAME : sub138
FIELD ID_REQUIRES OF STEP_COMSUM
IS A SUBSET OF
FIELD ID OF STEP

CONSTRAINT NAME : sub139
FIELD ID_IS_REQUIRED OF STEP_EQUIP
IS A SUBSET OF
FIELD ID OF EQUIP

CONSTRAINT NAME : sub140
FIELD ID_REQUIRES OF STEP_EQUIP
IS A SUBSET OF
FIELD ID OF STEP

CONSTRAINT NAME : sub141
FIELD ID_IS_DISPLAYED OF STEP_GRAPHIC
IS A SUBSET OF
FIELD ID OF GRAPHIC1

CONSTRAINT NAME : sub142
FIELD ID_DISPLAYS OF STEP_GRAPHIC
IS A SUBSET OF
FIELD ID OF STEP

CONSTRAINT NAME : sub143
FIELD ID_IS_REFERENCED OF STEP_NOTE
IS A SUBSET OF
FIELD ID OF NOTE

CONSTRAINT NAME : sub144
FIELD ID_REFERS OF STEP_NOTE
IS A SUBSET OF
FIELD ID OF STEP

CONSTRAINT NAME : sub145
FIELD ID_IS_REQUIRED OF STEP_PERSON
IS A SUBSET OF
FIELD ID OF PERSON

CONSTRAINT NAME : sub146
FIELD ID_REQUIRES OF STEP_PERSON
IS A SUBSET OF
FIELD ID OF STEP

CONSTRAINT NAME : sub147
FIELD ID_IS_INVOKED OF STEP_PROCESS
IS A SUBSET OF
FIELD ID OF PROCESS

CONSTRAINT NAME : sub148
FIELD ID_INVOKES OF STEP_PROCESS
IS A SUBSET OF
FIELD ID OF STEP

CONSTRAINT NAME : sub149
FIELD ID_IS_PRESENTED OF STEP_PROMPT
IS A SUBSET OF
FIELD ID OF PROMPT

CONSTRAINT NAME : sub150
FIELD ID_PPRESENTS OF STEP_PROMPT
IS A SUBSET OF
FIELD ID OF STEP

CONSTRAINT NAME : sub151
FIELD ID_TESTS OF STEP_REQCOND
IS A SUBSET OF
FIELD ID OF STEP

CONSTRAINT NAME : sub152
FIELD ID_IS_REFERENCED OF STEP_SUBSTEP
IS A SUBSET OF
FIELD ID OF STEP

CONSTRAINT NAME : sub153
FIELD ID_REFERS OF STEP_SUBSTEP

IS A SUBSET OF
 FIELD ID OF STEP

CONSTRAINT NAME : sub154
 FIELD ID_PPRESENTS OF STEP_TABLE
 IS A SUBSET OF
 FIELD ID OF STEP

CONSTRAINT NAME : sub155
 FIELD ID_IS_PRESENTED OF STEP_TABLE
 IS A SUBSET OF
 FIELD ID_CONTAINS OF TABLE_COLHDDEF

CONSTRAINT NAME : sub156
 FIELD ID_IS_PRESENTED OF STEP_TABLE
 IS A SUBSET OF
 FIELD ID_CONTAINS OF TABLE_ENTRY

CONSTRAINT NAME : sub157
 FIELD ID_USES OF STEP_VERB
 IS A SUBSET OF
 FIELD ID OF STEP

CONSTRAINT NAME : sub158
 FIELD ID_INVOKES OF STEP_VIDEO
 IS A SUBSET OF
 FIELD ID OF STEP

CONSTRAINT NAME : sub159
 FIELD ID_IS_INVOKED OF STEP_VIDEO
 IS A SUBSET OF
 FIELD ID OF VIDEO

CONSTRAINT NAME : sub160
 FIELD ID_REFERS OF STEP_WARNING
 IS A SUBSET OF
 FIELD ID OF STEP

CONSTRAINT NAME : sub161
 FIELD ID_IS_REFERENCED OF STEP_WARNING
 IS A SUBSET OF
 FIELD ID OF WARNING

CONSTRAINT NAME : sub162
 FIELD ID_IS_REFERENCED OF SYSTEM_DESCINFO
 IS A SUBSET OF
 FIELD ID OF DESCINFO

CONSTRAINT NAME : sub163
 FIELD ID_IS_REFERENCED OF SYSTEM_FAULTINF
 IS A SUBSET OF
 FIELD ID_IDENTIFIES OF FAULTINF_FAULT

CONSTRAINT NAME : sub164
FIELD ID_IS_REFERENCED OF SYSTEM_FAULTINF
IS A SUBSET OF
FIELD ID_IDENTIFIES OF FAULTINF_TEST

CONSTRAINT NAME : sub165
FIELD ID_IS_IDENTIFIED OF SYSTEM_PARTINFO
IS A SUBSET OF
FIELD ID OF PARTINFO

CONSTRAINT NAME : sub166
FIELD ID_IS_REFERENCED OF SYSTEM_TASK
IS A SUBSET OF
FIELD ID OF TASK

CONSTRAINT NAME : sub167
FIELD ID_CONTAINS OF TABLE_COLHDDEF
IS A SUBSET OF
FIELD ID OF COMMON_ELEMENT

CONSTRAINT NAME : sub168
FIELD ID_CONTAINS OF TABLE_ENTRY
IS A SUBSET OF
FIELD ID OF COMMON_ELEMENT

CONSTRAINT NAME : sub169
FIELD ID_IS_CONTAINED OF TABLE_COLHDDEF
IS A SUBSET OF
FIELD ID OF COLHDDEF

CONSTRAINT NAME : sub170
FIELD ID_IS_CONTAINED OF TABLE_ENTRY
IS A SUBSET OF
FIELD ID OF ENTRY

CONSTRAINT NAME : sub171
FIELD ID OF TASK
IS A SUBSET OF
FIELD ID OF COMMON_ELEMENT

CONSTRAINT NAME : sub172
FIELD ID_IS_REFERENCED OF TASK_CAUTION
IS A SUBSET OF
FIELD ID OF CAUTION

CONSTRAINT NAME : sub173
FIELD ID_REFERS OF TASK_CAUTION
IS A SUBSET OF
FIELD ID OF TASK

CONSTRAINT NAME : sub174

FIELD ID_IS_REQUIRED OF TASK_CONSUM
 IS A SUBSET OF
 FIELD ID OF CONSUM

CONSTRAINT NAME : sub175
 FIELD ID_REQUIRES OF TASK_CONSUM
 IS A SUBSET OF
 FIELD ID OF TASK

CONSTRAINT NAME : sub176
 FIELD ID_IS_REQUIRED OF TASK_EQUIP
 IS A SUBSET OF
 FIELD ID OF EQUIP

CONSTRAINT NAME : sub177
 FIELD ID_REQUIRES OF TASK_EQUIP
 IS A SUBSET OF
 FIELD ID OF TASK

CONSTRAINT NAME : sub178
 FIELD ID_FOLLOWED OF TASK_FOLLOWON
 IS A SUBSET OF
 FIELD ID OF TASK

CONSTRAINT NAME : sub179
 FIELD ID_IS_REFERENCED OF TASK_NOTE
 IS A SUBSET OF
 FIELD ID OF NOTE

CONSTRAINT NAME : sub180
 FIELD ID_REFERS OF TASK_NOTE
 IS A SUBSET OF
 FIELD ID OF TASK

CONSTRAINT NAME : sub181
 FIELD ID_IS_REQUIRED OF TASK_PERSON
 IS A SUBSET OF
 FIELD ID OF PERSON

CONSTRAINT NAME : sub182
 FIELD ID_REQUIRES OF TASK_PERSON
 IS A SUBSET OF
 FIELD ID OF TASK

CONSTRAINT NAME : sub183
 FIELD ID_TESTS OF TASK_REQCOND
 IS A SUBSET OF
 FIELD ID OF TASK

CONSTRAINT NAME : sub184
 FIELD ID_IS_REFERENCED OF TASK_STEP
 IS A SUBSET OF

FIELD ID OF STEP

CONSTRAINT NAME : sub185
FIELD ID_REFERS OF TASK_STEP
IS A SUBSET OF
FIELD ID OF TASK

CONSTRAINT NAME : sub186
FIELD ID_USES OF TASK_VERB
IS A SUBSET OF
FIELD ID OF TASK

CONSTRAINT NAME : sub187
FIELD ID_REFERS OF TASK_WARNING
IS A SUBSET OF
FIELD ID OF TASK

CONSTRAINT NAME : sub188
FIELD ID_IS_REFERENCED OF TASK_WARNING
IS A SUBSET OF
FIELD ID OF WARNING

CONSTRAINT NAME : sub189
FIELD ID OF TEST
IS A SUBSET OF
FIELD ID OF COMMON_ELEMENT

CONSTRAINT NAME : sub190
FIELD ID_IS_DISPLAYED OF TEST
IS A SUBSET OF
FIELD ID OF TEXT

CONSTRAINT NAME : sub191
FIELD ID_IS_PRODUCED OF TEST_OUTCOME
IS A SUBSET OF
FIELD ID OF OUTCOME

CONSTRAINT NAME : sub192
FIELD ID_PRODUCES OF TEST_OUTCOME
IS A SUBSET OF
FIELD ID OF TEST

CONSTRAINT NAME : sub193
FIELD ID_IS_IDENTIFIED OF TEST_TASK
IS A SUBSET OF
FIELD ID OF TASK

CONSTRAINT NAME : sub194
FIELD ID_IDENTIFIES OF TEST_TASK
IS A SUBSET OF
FIELD ID OF TEST

CONSTRAINT NAME : sub195
FIELD ID OF TEXT
IS A SUBSET OF
FIELD ID OF COMMON_ELEMENT

CONSTRAINT NAME : sub196
FIELD ATT_VALUE_IS_CONTAINED OF TEXT
IS A SUBSET OF
FIELD ATT_VALUE OF ATTVALUE

CONSTRAINT NAME : sub197
FIELD ID_IS_DISPLAYED OF VALUE
IS A SUBSET OF
FIELD ID OF TEXT

CONSTRAINT NAME : sub198
FIELD ID OF VIDEO
IS A SUBSET OF
FIELD ID OF COMMON_ELEMENT

CONSTRAINT NAME : sub199
FIELD ID OF WARNING
IS A SUBSET OF
FIELD ID OF COMMON_ELEMENT

CONSTRAINT NAME : sub200
FIELD ID_IS_DISPLAYED OF WARNING
IS A SUBSET OF
FIELD ID OF TEXT

CONSTRAINT NAME : sub201
FIELD ID_HAS OF XREF_ATTNAME
IS A SUBSET OF
FIELD ID OF XREF

(* Appendix I - CDM/Express Data Model

90 CDM/Express Data Model *)

SCHEMA content_data_model;

EXPORT EVERYTHING;

TYPE action_code = string(5);
END_TYPE;

TYPE agent_id = string(7);
END_TYPE;

TYPE atname = string(15);
END_TYPE;

TYPE att_value = string(10);
END_TYPE;

TYPE cage_code = string(20);
END_TYPE;

TYPE coding_code = string(80);
END_TYPE;

TYPE col_number = real(4);
END_TYPE;

TYPE colhddef_name = string(20);
END_TYPE;

TYPE colnum_number = real(4);
END_TYPE;

TYPE config_data = string(20);
END_TYPE;

TYPE default_data = string(80);
END_TYPE;

TYPE exref_id = string(10);
END_TYPE;

TYPE file_data = string(80);
END_TYPE;

TYPE govstd_code = string(20);
END_TYPE;

TYPE hci_code = string(10);
END_TYPE;

```
TYPE id = string(10);  
END_TYPE;  
  
TYPE indx_number = string(15);  
END_TYPE;  
  
TYPE itemid = string(10);  
END_TYPE;  
  
TYPE list_name = string(20);  
END_TYPE;  
  
TYPE maint_level = string(10);  
END_TYPE;  
  
TYPE mfg_code = string(20);  
END_TYPE;  
  
TYPE milspec_code = string(20);  
END_TYPE;  
  
TYPE minsize_code = string(10);  
END_TYPE;  
  
TYPE mtbf_time = real(10);  
END_TYPE;  
  
TYPE name = string(20);  
END_TYPE;  
  
TYPE nounid = string(80);  
END_TYPE;  
  
TYPE nountype = string(80);  
END_TYPE;  
  
TYPE nsn = string(20);  
END_TYPE;  
  
TYPE number = real(10);  
END_TYPE;  
  
TYPE op_data = string(3);  
END_TYPE;  
  
TYPE partnum = string(80);  
END_TYPE;  
  
TYPE pctext = string(9999);  
END_TYPE;
```

```

TYPE penpatt_data = string(10);
END_TYPE;

TYPE penshape_data = string(10);
END_TYPE;

TYPE polarity_data = string(3);
END_TYPE;

TYPE range_data = string(80);
END_TYPE;

TYPE refdes = string(10);
END_TYPE;

TYPE refid = string(10);
END_TYPE;

TYPE relation_data = string(10);
END_TYPE;

TYPE release_code = string(10);
END_TYPE;

TYPE rep_level = string(10);
END_TYPE;

TYPE row_number = real(10);
END_TYPE;

TYPE security_code = string(10);
END_TYPE;

TYPE select_data = string(8);
END_TYPE;

TYPE smr_code = string(6);
END_TYPE;

TYPE start_data = string(15);
END_TYPE;

TYPE stop_data = string(15);
END_TYPE;

TYPE time = real(10);
END_TYPE;

TYPE track_data = string(10);
END_TYPE;

TYPE transf_data = real(9);

```

```

END_TYPE;

TYPE type_data = string(80);
END_TYPE;

TYPE unitsper_number = string(10);
END_TYPE;

TYPE uom_unit = string(15);
END_TYPE;

TYPE usablon_code = string(15);
END_TYPE;

TYPE user_id = string(10);
END_TYPE;

TYPE valstat_data = string(10);
END_TYPE;

TYPE version_data = string(10);
END_TYPE;

TYPE verstat_data = string(10);
END_TYPE;

TYPE weight_factor = real(10);
END_TYPE;

TYPE window_data = string(10);
END_TYPE;

ENTITY action;
    with_action_code          : action_code;
    UNIQUE
    with_action_code;
END_ENTITY;

ENTITY agent;
    with_agent_id              : agent_id;
    UNIQUE
    with_agent_id;
END_ENTITY;

ENTITY annot
    SUBTYPE OF (common_element);
    annotated_by_user          : OPTIONAL user;
    displays_text              : text;
END_ENTITY;

ENTITY assertion
    SUBTYPE OF (cdm_element);

```

```

    asserts_property      : property;
    asserts_value         : SET [1:#] OF value;
END_ENTITY;

```

```

ENTITY attname;
    with_atname           : atname;
UNIQUE
    with_atname;
END_ENTITY;

```

```

ENTITY attvalue;
    with_att_value        : att_value;
    refers_elmntref        : elmntref;
    has_attname            : attname;
UNIQUE
    with_att_value;
END_ENTITY;

```

```

ENTITY audio
    SUBTYPE OF (common_element);
    refers_to_file         : OPTIONAL file;
    refers_to_exrefid      : OPTIONAL exrefid;
END_ENTITY;

```

```

ENTITY cage;
    with_cage_code         : cage_code;
UNIQUE
    with_cage_code;
END_ENTITY;

```

```

ENTITY caution
    SUBTYPE OF (common_element);
    displays_text          : text;
END_ENTITY;

```

```

ENTITY cdm_element
    SUPERTYPE OF ((xref OR
                    value OR
                    property OR
                    precondition OR
                    partbase OR
                    list OR
                    entry OR
                    context OR
                    common_element OR
                    colhddef OR
                    choice OR
                    assertion) XOR
                    NULL);

```

```

    with_id                : id;
UNIQUE
    with_id;

```



```

END_ENTITY;

ENTITY choice
  SUBTYPE OF (cdm_element);
  contains_value      : SET [1:#] OF value;
  displays_text       : text;
END_ENTITY;

ENTITY coding;
  with_coding_code    : coding_code;
UNIQUE
  with_coding_code;
END_ENTITY;

ENTITY col;
  with_col_number     : col_number;
UNIQUE
  with_col_number;
END_ENTITY;

ENTITY colhdddef
  SUBTYPE OF (cdm_element);
  with_colhdddef_name : colhdddef_name;
  contains_type       : OPTIONAL type;
  contains_colnum     : colnum;
UNIQUE
  with_colhdddef_name;
END_ENTITY;

ENTITY colnum;
  with_colnum_number  : colnum_number;
UNIQUE
  with_colnum_number;
END_ENTITY;

ENTITY common_element
  SUPERTYPE OF ((warning OR
                 video OR
                 verb OR
                 text OR
                 test OR
                 techinfo OR
                 task OR
                 table OR
                 system OR
                 step OR
                 reqcond OR
                 rect OR
                 prompt OR
                 process OR
                 person OR
                 partinfo OR

```

```

outcome OR
operinfo OR
note OR
menu OR
grphprim OR
graphic OR
fltstate OR
fillin OR
faultinf OR
fault OR
equip OR
dictitem OR
descinfo OR
consum OR
caution OR
audio OR
annot) XOR
NULL)
SUBTYPE OF (cdm_element);
with_refid          : refid;
with_name           : OPTIONAL name;
with_itemid         : OPTIONAL itemid;
refers_to_xref      : OPTIONAL SET [1:#] OF xref;
contains_type       : OPTIONAL type;
contains_context    : context;
END_ENTITY;

ENTITY config;
  with_config_data   : config_data;
UNIQUE
  with_config_data;
END_ENTITY;

ENTITY consum
  SUBTYPE OF (common_element);
  measured_by_uom    : uom;
  has_qty            : qty;
  is_specified_by_milspec : milspec;
  is_specified_by_mfgcode : mfgcode;
  is_specified_by_govstd  : govstd;
END_ENTITY;

ENTITY context
  SUBTYPE OF (cdm_element);
  contains_verstat    : OPTIONAL verstat;
  contains_version    : OPTIONAL SET [1:#] OF version;
  contains_valstat    : OPTIONAL valstat;
  contains_track      : OPTIONAL SET [1:#] OF track;
  contains_security   : OPTIONAL security;
  contains_release    : OPTIONAL SET [1:#] OF release;
  tests_precond       : OPTIONAL SET [1:#] OF precondition;
  contains_maintlvl   : OPTIONAL maintlvl;

```

```

    contains_config          : OPTIONAL SET [1:#] OF config;
END_ENTITY;

ENTITY default;
    with_default_data       : default_data;
UNIQUE
    with_default_data;
END_ENTITY;

ENTITY descinfo
    SUBTYPE OF (common_element);
    invokes_video           : OPTIONAL SET [1:#] OF video;
    displays_text           : OPTIONAL text;
    presents_table          : OPTIONAL SET [1:#] OF table;
    presents_prompt         : OPTIONAL SET [1:#] OF prompt;
    invokes_process         : OPTIONAL SET [1:#] OF process;
    displays_graphic        : OPTIONAL SET [1:#] OF graphic;
    refers_to_descinfo      : OPTIONAL SET [1:#] OF descinfo;
    invokes_audio           : OPTIONAL SET [1:#] OF audio;
    asserts_assertion       : OPTIONAL SET [1:#] OF assertion;
    refers_to_annot         : OPTIONAL SET [1:#] OF annot;
END_ENTITY;

ENTITY dictitem
    SUBTYPE OF (common_element);
    refers_to_elmntref      : SET [1:#] OF elmntref;
END_ENTITY;

ENTITY elmntref;
    same_as_cdm_element     : cdm_element;
UNIQUE
    same_as_cdm_element;
END_ENTITY;

ENTITY entry
    SUBTYPE OF (cdm_element);
    displays_text           : OPTIONAL text;
    contains_row            : row;
    refers_to_elmntref      : OPTIONAL elmntref;
    contains_col            : col;
END_ENTITY;

ENTITY equip
    SUBTYPE OF (common_element);
    has_qty                 : qty;
    has_alt_equip           : OPTIONAL SET [1:#] OF equip;
END_ENTITY;

ENTITY esttime;
    with_time               : time;
UNIQUE
    with_time;

```

END_ENTITY;

ENTITY exrefid;
 with_exref_id : exref_id;
UNIQUE
 with_exref_id;
END_ENTITY;

ENTITY fault
 SUBTYPE OF (common_element);
 identifies_rect : SET [1:#] OF rect;
 exculpating_fltstate : OPTIONAL SET [1:#] OF fltstate;
 displays_text : OPTIONAL text;
 points_to_partinfo : OPTIONAL SET [1:#] OF partinfo;
 has_mtbf : OPTIONAL mtbf;
END_ENTITY;

ENTITY faultinf
 SUBTYPE OF (common_element);
 identifies_test : SET [1:#] OF test;
 identifies_fault : SET [1:#] OF fault;
END_ENTITY;

ENTITY file;
 with_file_data : file_data;
UNIQUE
 with_file_data;
END_ENTITY;

ENTITY fillin
 SUBTYPE OF (common_element);
 displays_text : text;
 contains_range : OPTIONAL range;
 asserts_property : property;
 has_default : OPTIONAL default;
END_ENTITY;

ENTITY fltstate
 SUBTYPE OF (common_element);
 scaled_by_weight : OPTIONAL SET [1:#] OF weight;
 displays_text : OPTIONAL text;
 identifies_test : OPTIONAL test;
 implicated_by_fault : OPTIONAL SET [1:#] OF fault;
 exculpated_by_fault : OPTIONAL SET [1:#] OF fault;
END_ENTITY;

ENTITY followon;
 with_refid : refid;
UNIQUE
 with_refid;
END_ENTITY;

```

ENTITY govstd;
  with_govstd_code      : govstd_code;
UNIQUE
  with_govstd_code;
END_ENTITY;

ENTITY graphic
  SUBTYPE OF (common_element);
  specifies_window      : OPTIONAL SET [1:#] OF window;
  specifies_transform    : OPTIONAL SET [1:#] OF transform;
  displays_text         : OPTIONAL text;
  specifies_penshape     : OPTIONAL penshape;
  specifies_penpatt      : OPTIONAL penpatt;
  specifies_minsize     : OPTIONAL SET [1:#] OF minsize;
  invokes_grphprim      : SET [1:#] OF grphprim;
  specifies_focus_of_graphic : OPTIONAL SET [1:#] OF graphic;
END_ENTITY;

ENTITY grphprim
  SUBTYPE OF (common_element);
  specifies_window      : OPTIONAL SET [1:#] OF window;
  specifies_transform    : OPTIONAL SET [1:#] OF transform;
  displays_text         : OPTIONAL text;
  stopping_at_stop      : OPTIONAL stop;
  starting_at_start     : OPTIONAL start;
  specifies_penshape     : OPTIONAL penshape;
  specifies_penpatt      : OPTIONAL penpatt;
  contains_pcddata      : OPTIONAL pcddata;
  specifies_minsize     : OPTIONAL SET [1:#] OF minsize;
  refers_to_file        : OPTIONAL file;
  specifies_coding       : OPTIONAL coding;
END_ENTITY;

ENTITY hci;
  with_hci_code         : hci_code;
UNIQUE
  with_hci_code;
END_ENTITY;

ENTITY indxnum;
  with_indx_number      : indx_number;
UNIQUE
  with_indx_number;
END_ENTITY;

ENTITY list
  SUBTYPE OF (cdm_element);
  with_refid            : refid;
  with_list_name        : OPTIONAL list_name;
  refers_to_elmntref    : SET [1:#] OF elmntref;
  contains_context      : context;
END_ENTITY;

```

```

ENTITY maintlvl;
  with_maint_level      : maint_level;
UNIQUE
  with_maint_level;
END_ENTITY;

ENTITY menu
  SUBTYPE OF (common_element);
  has_select            : OPTIONAL select;
  asserts_property     : property;
  has_default          : OPTIONAL SET [1:#] OF default;
  has_choice           : SET [1:#] OF choice;
  displays_text        : text;
END_ENTITY;

ENTITY mfgcode;
  with_mfg_code         : mfg_code;
UNIQUE
  with_mfg_code;
END_ENTITY;

ENTITY milspec;
  with_milspec_code     : milspec_code;
UNIQUE
  with_milspec_code;
END_ENTITY;

ENTITY minsize;
  with_minsize_code     : minsize_code;
UNIQUE
  with_minsize_code;
END_ENTITY;

ENTITY mtbf;
  with_mtbf_time        : mtbf_time;
UNIQUE
  with_mtbf_time;
END_ENTITY;

ENTITY note
  SUBTYPE OF (common_element);
  displays_text         : text;
END_ENTITY;

ENTITY op;
  with_op_data          : op_data;
UNIQUE
  with_op_data;
END_ENTITY;

ENTITY operinfo

```

```

SUBTYPE OF (common_element);
  refers_to_task      : OPTIONAL SET [1:#] OF task;
  refers_to_descinfo  : OPTIONAL SET [1:#] OF descinfo;
END_ENTITY;

```

```

ENTITY outcome
  SUBTYPE OF (common_element);
  displays_text      : OPTIONAL text;
  tests_precond      : SET [1:#] OF precondition;
  identifies_fltstate : SET [1:#] OF ftest;
END_ENTITY;

```

```

ENTITY partbase
  SUBTYPE OF (cdm_element);
  with_refid      : refid;
  with_partnum    : partnum;
  with_nsn        : nsn;
  with_name       : OPTIONAL name;
  with_itemid     : OPTIONAL itemid;
  refers_xref     : OPTIONAL SET [1:#] OF xref;
  contains_type   : OPTIONAL type;
  contains_smr    : smr;
  contains_hci    : hci;
  made_by_cage    : cage;
  UNIQUE
    with_nsn;
    with_name;
END_ENTITY;

```

```

ENTITY partinfo
  SUBTYPE OF (common_element);
  with_refdes      : refdes;
  with_nounid      : OPTIONAL nounid;
  with_nountype    : OPTIONAL nountype;
  is_identified_by_usablon : usablon;
  requires_unitsper : unitsper;
  identifies_replvl : OPTIONAL replvl;
  identifies_partbase : SET [1:#] OF partbase;
  has_mtbfd       : mtbf;
  is_referenced_by_indxnum : indxnum;
  identifies_graphic : SET [1:#] OF graphic;
  UNIQUE
    with_refdes;
END_ENTITY;

```

```

ENTITY pcddata;
  with_pctext      : pctext;
  UNIQUE
    with_pctext;
END_ENTITY;

```

```

ENTITY penpatt;

```

```

    with_penpatt_data      : penpatt_data;
UNIQUE
    with_penpatt_data;
END_ENTITY;

ENTITY penshape;
    with_penshape_data      : penshape_data;
UNIQUE
    with_penshape_data;
END_ENTITY;

ENTITY person
    SUBTYPE OF (common_element);
    has_qty                  : qty;
END_ENTITY;

ENTITY polarity;
    with_polarity_data      : polarity_data;
UNIQUE
    with_polarity_data;
END_ENTITY;

ENTITY precon
    SUBTYPE OF (cdm_element);
    tests_value              : SET [1:#] OF value;
    tests_property           : property;
    contains_polarity        : OPTIONAL polarity;
    contains_op              : OPTIONAL op;
END_ENTITY;

ENTITY process
    SUBTYPE OF (common_element);
    refers_to_file           : OPTIONAL file;
    refers_to_exrefid        : OPTIONAL exrefid;
END_ENTITY;

ENTITY prompt
    SUBTYPE OF (common_element);
    displays_text            : OPTIONAL text;
    calls_menu               : OPTIONAL SET [1:#] OF menu;
    contains_fillin          : OPTIONAL SET [1:#] OF fillin;
END_ENTITY;

ENTITY property
    SUBTYPE OF (cdm_element);
    displays_text            : text;
    refers_to_elmntref       : OPTIONAL elmntref;
END_ENTITY;

ENTITY qty;
    with_number              : number;
UNIQUE

```


with_number;
END_ENTITY;

ENTITY range;
with_range_data : range_data;
UNIQUE
with_range_data;
END_ENTITY;

ENTITY rect
SUBTYPE OF (common_element);
displays_text : OPTIONAL text;
identifies_test : OPTIONAL SET [1:#] OF test;
identifies_task : SET [1:#] OF task;
is_performed_by_agent : OPTIONAL agent;
identifies_action : OPTIONAL action;
END_ENTITY;

ENTITY relation;
with_relation_data : relation_data;
UNIQUE
with_relation_data;
END_ENTITY;

ENTITY release;
with_release_code : release_code;
UNIQUE
with_release_code;
END_ENTITY;

ENTITY replvl;
with_rep_level : rep_level;
UNIQUE
with_rep_level;
END_ENTITY;

ENTITY reqcond
SUBTYPE OF (common_element);
tests_precond : OPTIONAL SET [1:#] OF precondition;
refers_to_elmntref : OPTIONAL SET [1:#] OF elmntref;
END_ENTITY;

ENTITY row;
with_row_number : row_number;
UNIQUE
with_row_number;
END_ENTITY;

ENTITY security;
with_security_code : security_code;
UNIQUE
with_security_code;

END_ENTITY;

ENTITY select;
 with_select_data : select_data;
UNIQUE
 with_select_data;
END_ENTITY;

ENTITY smr;
 with_smr_code : smr_code;
UNIQUE
 with_smr_code;
END_ENTITY;

ENTITY start;
 with_start_data : start_data;
UNIQUE
 with_start_data;
END_ENTITY;

ENTITY step
 SUBTYPE OF (common_element);
 refers_to_warning : OPTIONAL SET [1:#] OF warning;
 invokes_video : OPTIONAL SET [1:#] OF video;
 uses_verb : OPTIONAL SET [1:#] OF verb;
 displays_text : text;
 presents_table : OPTIONAL SET [1:#] OF table;
 refers_to_step : OPTIONAL SET [1:#] OF step;
 tests_reqcond : OPTIONAL SET [1:#] OF reqcond;
 presents_prompt : OPTIONAL SET [1:#] OF prompt;
 invokes_process : OPTIONAL SET [1:#] OF process;
 requires_person : OPTIONAL SET [1:#] OF person;
 refers_to_note : OPTIONAL SET [1:#] OF note;
 displays_graphic : OPTIONAL SET [1:#] OF graphic;
 requires_esttime : OPTIONAL esttime;
 requires_equip : OPTIONAL SET [1:#] OF equip;
 requires_consum : OPTIONAL SET [1:#] OF consum;
 refers_to_caution : OPTIONAL SET [1:#] OF caution;
 invokes_audio : OPTIONAL SET [1:#] OF audio;
 asserts_assertion : OPTIONAL SET [1:#] OF assertion;
 contains_annot : OPTIONAL SET [1:#] OF annot;
END_ENTITY;

ENTITY stop;
 with_stop_data : stop_data;
UNIQUE
 with_stop_data;
END_ENTITY;

ENTITY system
 SUBTYPE OF (common_element);
 refers_to_task : OPTIONAL SET [1:#] OF task;

```

    refers_to_system      : OPTIONAL SET [1:#] OF system;
    identifies_partinfo   : OPTIONAL SET [1:#] OF partinfo;
    refers_to_operinfo    : OPTIONAL SET [1:#] OF operinfo;
    refers_to_faultinf    : OPTIONAL SET [1:#] OF faultinf;
    refers_to_descinfo    : OPTIONAL SET [1:#] OF descinfo;
END_ENTITY;

```

ENTITY table

```

    SUBTYPE OF (common_element);
    contains_entry        : SET [1:#] OF entry;
    contains_colhddef     : SET [1:#] OF colhddef;
END_ENTITY;

```

ENTITY task

```

    SUBTYPE OF (common_element);
    refers_to_warning     : OPTIONAL SET [1:#] OF warning;
    refers_to_step        : SET [1:#] OF step;
    tests_reqcond         : OPTIONAL SET [1:#] OF reqcond;
    requires_person       : OPTIONAL SET [1:#] OF person;
    refers_to_note        : OPTIONAL SET [1:#] OF note;
    followed_by_followon  : OPTIONAL SET [1:#] OF followon;
    requires_esttime      : OPTIONAL esttime;
    requires_equip        : OPTIONAL SET [1:#] OF equip;
    requires_consum       : OPTIONAL SET [1:#] OF consum;
    refers_to_caution    : OPTIONAL SET [1:#] OF caution;
END_ENTITY;

```

ENTITY techinfo

```

    SUBTYPE OF (common_element);
    contains_system       : OPTIONAL SET [1:#] OF system;
END_ENTITY;

```

ENTITY test

```

    SUBTYPE OF (common_element);
    displays_text         : OPTIONAL text;
    identifies_task       : SET [1:#] OF task;
    is_represented_by_range : OPTIONAL range;
    produces_outcome      : SET [1:#] OF outcome;
    is_performed_by_agent : OPTIONAL agent;
END_ENTITY;

```

ENTITY text

```

    SUBTYPE OF (common_element);
    contains_pdata        : OPTIONAL pdata;
    contains_attvalue     : OPTIONAL attvalue;
END_ENTITY;

```

ENTITY track;

```

    with_track_data      : track_data;
    UNIQUE
    with_track_data;
END_ENTITY;

```

```

ENTITY transform;
  with_transf_data      : transf_data;
UNIQUE
  with_transf_data;
END_ENTITY;

ENTITY type;
  with_type_data        : type_data;
UNIQUE
  with_type_data;
END_ENTITY;

ENTITY unitsper;
  with_unitsper_number  : unitsper_number;
UNIQUE
  with_unitsper_number;
END_ENTITY;

ENTITY uom;
  with_uom_unit         : uom_unit;
UNIQUE
  with_uom_unit;
END_ENTITY;

ENTITY usablon;
  with_usablon_code     : usablon_code;
UNIQUE
  with_usablon_code;
END_ENTITY;

ENTITY user;
  with_user_id         : user_id;
UNIQUE
  with_user_id;
END_ENTITY;

ENTITY valstat;
  with_valstat_data     : valstat_data;
UNIQUE
  with_valstat_data;
END_ENTITY;

ENTITY value
  SUBTYPE OF (cdm_element);
  displays_text         : text;
END_ENTITY;

ENTITY verb
  SUBTYPE OF (common_element);
  is_used_by_task      : OPTIONAL SET [1:#] OF task;
END_ENTITY;

```

```

ENTITY version;
  with_version_data      : version_data;
  UNIQUE
    with_version_data;
END_ENTITY;

ENTITY verstat;
  with_verstat_data      : verstat_data;
  UNIQUE
    with_verstat_data;
END_ENTITY;

ENTITY video
  SUBTYPE OF (common_element);
  refers_to_file          : OPTIONAL file;
  refers_to_exrefid       : OPTIONAL exrefid;
END_ENTITY;

ENTITY warning
  SUBTYPE OF (common_element);
  displays_text           : text;
END_ENTITY;

ENTITY weight;
  with_weight_factor      : weight_factor;
  UNIQUE
    with_weight_factor;
END_ENTITY;

ENTITY window;
  with_window_data        : window_data;
  UNIQUE
    with_window_data;
END_ENTITY;

ENTITY xref
  SUBTYPE OF (cdm_element);
  contains_relation       : OPTIONAL relation;
  refers_to_exrefid       : OPTIONAL exrefid;
  refers_to_elmntref      : OPTIONAL elmntref;
  has_attname             : OPTIONAL attname;
END_ENTITY;

END_SCHEMA; -- end of content_data_model

```

Appendix J - Cross Reference of CDM and MIL-STD-1388-2B Data Elements

100 Cross Reference of CDM and MIL-STD-1388-2B Data Elements

The following table shows that the data of CDM data elements can be derived from one or more data elements of MIL-STD-1388-2B.

CDM		MIL-STD-1388-2B		
No.	Data Element/Attribute	No.	DE Code	Data Element Title
070	system/name	093	EIACODXA	End Item Acronym Code
133	type	194	LCNINDXB	LCN Indenture Code
056	itemid	193	LSACONXB	LSA Control Number
		019	ALTLCNXB	Alternate LCN Code
		197	LCNTYPXB	LCN Type
		495	UOCSEIXC	Usable on Code
032	descinfo	334	RAMCNABB	RAM Narrative Code
		176	RAMNARBB	RAM Narrative
124	task	193	TLSACNBH	Task Requirement LCN
		019	ALTLCNBH	Alternate LCN Code
		197	LCNTYPBH	LCN Type
		419	TTASKCBH	Task Code
		425	TATYPEBH	Task Type
070	task/name	423	TASKIDCA	Task Identification
133	type	419	TASKCDCA	Task Code (1st Char.)
056	itemid	193	LSACONXB	LSA Control Number
		019	ALTLCNXB	Alternate LCN Code
		197	LCNTYPXB	LCN Type
		419	TASKCDCA	Task Code
037	esttime	218	PRDMMHCA	Predicted Mean Man-hours
070	person/name	281	SUBPIDCD	Subtask Person ID
133	type	379	SKSPCDGA	Skill Specialty Code
056	itemid	193	LSACONXB	LSA Control Number
		019	ALTLCNXB	Alternate LCN Code
		197	LCNTYPXB	LCN Type
		419	TASKCDCA	Task Code
		399	SUBNUMCB	Subtask Task Number
096	qty	281	SUBPIDCD	From "Subtask Person Id"
070	step/name	423	TASKIDCA	Task Identification
133	type	419	TASKCDCA	Task Code (1st char.)
056	itemid	193	LSACONXB	LSA Control Number
		019	ALTLCNXB	Alternate LCN Code
		197	LCNTYPXB	LCN Type
		419	TASKCDCA	Task Code
		399	SUBNUMCB	Subtask Number
037	esttime	220	SBMMETCB	Mean Minute Elapse Time

CDM		MIL-STD-1388-2B		
No.	Data Element/Attribute	No.	DE Code	Data Element Title
127	text	364	SUBNARCC	Subtask Description
152	warning	445	TEXSEQCC	Sequential Text Code
014	caution	364	SUBNARCC	Subtask Description
071	note	364	SUBNARCC	Subtask Description
070	equip/name	330	TSREFNCG	Task Support Ref Number
133	type	046	TSCAGECG	Task Support CAGE Code
056	itemid	173	SEICCDEA	Item Category Code
		193	LSACONXB	LSA Control Number
		019	ALTLCNXB	Alternate LCN Code
		197	LCNTYPXB	LCN Type
		419	TASKCDCA	Task Code
096	qty	312	SQTYTKCG	Quantity Per Task
		485	SQTKUMCG	Unit of Measure
036	equip	246	ALTFSCEH	Alternate Equip NSN FSC
		246	ALTNIIEH	Alternate Equip NSN NIIN
070	consum/name	178	ITNAMEHA	Item Name
133	type	173	ITMCATHG	Item Category Code
056	itemid	193	TSKLCNCI	LSA Control Number
		019	ALTLCNCI	Alternate LCN Code
		197	LCNTYPCI	LCN Type
		419	TASKCDCI	Task Code
063	mfgcode	330	REFNUMHA	Reference Number
		046	CAGECDHX	CAGE Code
096	qty	312	PQTYTKCI	Quantity Per Task
137	uom	485	PQTKUMCI	Unit of Measure
070	partinfo/name	178	ITMCATHG	Item Name
133	type	173	ITMCATHG	Item Category Code
056	itemid	193	PROLCNCI	Task Provisioning LCN
		019	ALTLCNXB	Alternate LCN Code
		197	LCNTYPXB	LCN Type
		419	TASKCDCA	Task Code
135	unitsper	309	QTYASYHG	Quantity Per Assembly
055	indxnum	158	INDCODHG	Indenture Code
140	usablon	495	UOCSEIXC	Usable On Code
068	mtbf	205	MRRONEHG	Maint. Repla. Rate I
		206	MRRTWOHG	Maint. Repla. Rate II
		207	MRRMODHG	Maint. Repla. Modifier
107	replvl	322	RMSSLIHG	Recom. Minimum Stock Lvl
070	partbase/name	178	ITNAMEHA	Item Name
133	type	173	ITMCATHG	Item Category Code

CDM		MIL-STD-1388-2B		
No.	Data Element/Attribute	No.	DE Code	Data Element Title
056	itemid	193	PROLCNCI	Task Provision LCN
		019	ALTLCNXB	Alternate LCN Code
		197	LCNTYPXB	LCN Type
		419	TASKCDCA	Task Code
082	partnum	330	REFNUMHA	Reference Number
C12	cage	046	CAGECDXH	CAGE Code
074	nsn	246	COGNSNHA	NSN Cognizance Code
074	nsn	246	SMMNSNHA	NSN Special Materiel ID
074	nsn	246	MATNSNHA	NSN Materiel Control CD
074	nsn	246	FSCNSNHA	NSN Classification
074	nsn	246	NIINSNHA	NSN National ID Number
074	nsn	246	ACTNSNHA	NSN Activity Code
115	smr	381	SMRCODHG	Sour., Maint. & Recover
051	hci	147	HARDCIHG	Hardness Critical Item
060	context/maintlvl	419		Task Code (Third Char.)

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix K - MIL-STD-1388-2B Data Elements Available for CDM Use

110 MIL-STD-1388-2B Data Elements Available for CDM Use

The following is a list of selected MIL-STD-1388-2B data elements pertinent to maintenance technical information useful for CDM. Some of the MIL-STD-1388-2B data elements listed below can be directly associated with CDM as shown in Appendix J. The other data elements listed below which can not be related to the current version of CDM should be considered with respect to the future expansion of CDM. For example, the source date for the calibration of the test equipment is available from LSAR. However, the current version of CDM is unable to use it.

The MIL-STD-1388-2B data elements are listed under their relational tables. The sequence of the relational tables shown below is that of Appendix A of MIL-STD-1388-2B.

XA - End Item Acronym Code Table

093	System/End Item Acronym Code
196	LCN Structure

XB - LSA Control Number Indentured Item Table

193	LSA Control Number (LCN)
019	Alternate LCN Code
197	LCN type
194	LCN Indenture Code
195	LCN Nomenclature
415	System/End Item Identifier

XC - System/End Item table

495	Usable on Code
175	System/End item Designator code

BB - RAM Characteristic Narrative table

334	RAM Characteristic Narrative Code
445	RAM Narrative Text Sequencing Code
176	RAM Characteristic Narrative

BH - Failure Mode Task Table

093	Task LCN
019	Task Alternate LCN Code
197	Task LCN Type
130	Task Failure Mode Indicator
419	Task Code
425	Task Type
202	Maintenance Interval
231	Maintenance Interval Measure. Base

CA - Task Requirement Table

419	Task Code
093	Referenced EIAC
193	Referenced LCN
019	Referenced Alternate LCN Code
419	Referenced Task Code
193	AOR LCN
019	AOR ALC
197	AOR LCN Type
231	Task AOR Measurement Base
423	Task Identification
422	Task Frequency
421	Task Criticality
148	Hardness Critical Procedure Code
151	Hazardous Maint. Procedure Code
289	Preventive Maint. Checks and Services Code
217	Measured/Predicted Mean Elapse Time
230	Primary/Secondary Means of Detection
350	Facility/Training Requirement Code
457	Training Rationale
350	Tool/Support Equipment Requirement Code
280	Task Performance Standards
420	Task Conditions

CB - Subtask Requirement Table

399	Subtask Number
423	Subtask Identification
193	Referenced Subtask LCN
019	Referenced Subtask Alternate LCN
197	Referenced Subtask LCN Type
399	Referenced Subtask Number
419	Referenced Subtask Task Code
220	Subtask Mean Minute Elapse Time
508	Subtask Work Area Code

CC - Sequential Subtask Description Table

445	Sequential Text Sequencing Code
364	Sequential Subtask Description

CD - Subtask Personnel Requirement Table

281	Subtask Personnel Identifier
379	Skill Specialty Code
219	Subtask Mean Man Minutes
380	Skill Specialty Evaluation Code

CE - Task Remark Table

341 Task Remark Reference Code
424 Task Remark

CG - Task Support Equipment table

330 Task Support Reference Number
046 Task Support CAGE Code
312 Support Item Per Task
485 Support Item Unit of Measure

CH - Task Manual Table

429 Technical Manual Code
432 Technical Manual Number
436 Technical Manual Volume
435 Technical Manual Section
433 Technical Manual Paragraph

EA - Support Equipment Table

046 Support CAGE Code
330 Support Equipment Reference No.
404 Support Equipment Full Item Name
173 Support Equipment Item Category Code
073 Calibration Internal
038 Calibration Item
040 Calibration Required
041 Calibration Standard
042 Calibration Time
035 Calibration Measurement Summary Recommend
220 Support Equipment MTBF
223 Support Equipment Mean Time Between Maintenance
261 Operating Height, Length, Width and Weight
381 Support Equipment SMR
434 Technical Manual Requirement Code
271 Operator Manual
379 Skill Specialty Code for Operator
362 Support Equipment Self Test Code
438 Test Language
441 Test point

EC - Support Equipment Parameters Table

039 Calibration procedure
277 Support Equipment Calibration Parameters

EE - Support Equipment Narrative Table

445 SE Narrative Text Sequencing Code
--- Support Equipment Narrative

EH - Alternate National Stock Number Table

246 Alternate Support Equipment NSN

UB - Unit Under Test and Test Support Equipment Table

193 UUT LCN
019 UUT Alternate LCN Code
197 UUT LCN type
330 Support Equipment Reference Number
046 Support Equipment Reference number
035 UUT Calibration Requirement Summary Code

UD - UUT Support Equipment Operational Test Program Table

330 Operational Test Program Reference Number
046 Operational test Program CAGE Code

UL - UUT Support Equipment Automatic Test Equipment Table

330 ATE Reference Number
046 ATE CAGE Number

HA - Provisioning Item Identification table

046 CAGE Code
330 reference Number
178 Item Name
179 Item code
246 National Stock Number
387 Special Material Content Code
384 Special Maintenance Item code
066 Criticality Code
490 Unit Length, Width and Height
491 Unit Weight

HG - Part Application Provisioning Table

409 Quantity per Assembly
306 Provisioning Technical Document. Selection Code
173 Item Category Code
097 Essentiality Code
381 SMR Code
205 Maintenance Replacement Code
322 Recomm. Minimum System Stock Level
017 Allowance Item Code
018 Allowance Item Quantity

HK - Parts Manual Description Table

432 Technical Manual Number
140 Figure Number
180 Item Number

Appendix L - CDM Data Elements Not Available From MIL-STD-1388-2B

120 CDM Data Elements Not Available From MIL-STD-1388-2B

The following is a list of CDM data elements for which the source data is not available from MIL-STD-1388-2B.

DED	Data Element Description
006	ASSERTION - Assertion
011	AUDIO - Audio Data
029	CONTEXT - Effectivity Analysis
041	FAULTINF - Fault Information Data
049	GRAPHIC - Graphic Element
061	MENU - Menu of Choices
094	PROMPT - User Input Prompt
108	REQCOND - Required Condition
123	TABLE - Tabular Information
151	VIDEO - Video Process

THIS PAGE INTENTIONALLY LEFT BLANK

Initial Distribution List

Copies

- 2 OASD (A&L)
 - 1 M. McGrath
 - 1 B. Lepisto
- 2 DSPO
 - 1 W. Gorham
 - 1 G. Desiderio
- 1 Army AMC, J. Lamb
- 1 AMC MRSA, E. Atkinson
- 1 AF MIO, Col. V. Ellis
- 10 AFHRL/LRC, D. Gunning
 - 1 AFLC/MMD, S. Holloway
 - 1 ASD/YFL, D. Mates
 - 2 AFSC/WRDC/MT
 - 1 L. Baker
 - 1 D. Rogash
 - 1 AFSC/AFTOMS, D. Henderson
 - 1 DLA, W. Simonson
 - 4 CNO, OP-403
 - 1 Capt. B. Hicks
 - 1 Cmd. M. Adams
 - 1 S. Brookins
 - 1 M. Weden
 - 1 SAPWAR, J. Penrod
 - 5 NAVSEA
 - 1 04 PA, R. Calogero
 - 1 04 TD, H. Felson
 - 1 04 PA, R. Di Geronimo
 - 1 507, D. Billingsley
 - 1 NAVSEALOGCENTER, W. Jones
 - 3 NAVSUP
 - 1 55, R. Houts
 - 1 551, P. Schmitt
 - 1 553, D. Kyle

Copies

- 2 NAVAIR
 - 1 4112, R. Golenbioski
 - 1 4114, S. McGill
- 1 Naval Postgraduate School Library
- 7 NIST
 - 1 D. Bettwy
 - 1 W. Danner
 - 1 A. Goldfine
 - 1 D. Jefferson
 - 1 M. Mitchell
 - 1 R. Morgan
 - 1 J. Tyler
- 8 CALS Industry Steering Group
 - 1 R. Longuemare
 - 1 H. Stormfeltz
 - 1 M. Plawsky (Logistic Process)
 - 1 J. Hays (LSA)
 - 1 R. Middleton (TM)
 - 1 J. Crawford (Standards)
 - 1 B. Kidwell (Infor. Techno.)
 - 1 L. Miller (AI)
- 3 IGES/PDES
 - 1 A. Day
 - 1 J. Wood (Tech. Publication)
 - 1 R. Bsharah (Product Life Cycle Support)
- 1 CALS Test Network, M. Lammers
- 1 PDES Inc., R. Kiggans
- 1 ANSI X3H4 IRDS, A. Winkler
- 1 IBM, M. Earl
- 1 SEI, G. Cooke
- 50 RJO, R. Shroder
- 2 Datalogics
 - 1 P. Gennusa
 - 1 J. Oldfield

Copies

1 General Dynamics, V. Botticelli

12 DTIC

77 DTRC

1	00	Capt. C. Graham
1	01	R. Metrey
1	12	G. Kerr
1	122	A. Camara
3	1223	J. Fuller
4	1226	J. Garner
40	1226	R. chen
1	1250	R. Schaffran
1	1251	J. Brainin
1	1290	R. Ploe
1	3411	C. Naas
20	3432	Reports Control
1	3421	Library (C)
1	3422	Library (A)

Additional copies of this report can be obtained from:

Reports Control Branch
Code 3432
David Taylor Research Center
Bethesda, MD 20084-5000

The FAX number for the Reports Control Branch is 301-227-3197.